Improving Visual Analysis of Streaming Networks

Jean Roberto Ponciano

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2020

Jean Roberto Ponciano

Improving Visual Analysis of Streaming Networks

Tese de doutorado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Bruno Augusto Nassif Travençolo
Coorientadora: Elaine Ribeiro de Faria Paiva

Uberlândia
2020

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

P795 2020

Ponciano, Jean Roberto, 1990-
Improving Visual Analysis of Streaming Networks [recurso eletrônico] / Jean Roberto Ponciano. - 2020.

Orientador: Bruno Augusto Nassif Travençolo.
Coorientadora: Elaine Ribeiro de Faria Paiva.
Tese (Doutorado) - Universidade Federal de Uberlândia, Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: http://doi.org/10.14393/ufu.te.2020.653
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Travençolo, Bruno Augusto Nassif, 1981-, (Orient.). II. Paiva, Elaine Ribeiro de Faria, 1980-, (Coorient.). III. Universidade Federal de Uberlândia. Pós-graduação em Ciência da Computação. IV. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:

Gizele Cristine Nunes do Couto - CRB6/2091

To all who helped me grow as a researcher, directly or not.

Acknowledgement

I thank Prof. Bruno A. N. Travençolo and Prof. Elaine R. Faria Paiva, my research supervisors, for their support and trust. Their guidance, patience, and advice were fundamental along the way. I also thank Prof. Luis E. C. Rocha and Prof. José Gustavo S. Paiva for the collaboration. Their valuable ideas, comments, and suggestions helped to improve this thesis.

I thank my parents Idalena and Giovani, and my sister Adrielly, for the support and patience they had over the past decade or more, since I started my undergraduate studies. I am also deeply grateful to my girlfriend Mariana Melo, a person that met me when I was passing through difficult times in my Ph.D. and chose to be on my side in every step since then, always encouraging and supporting me. This Ph.D. degree is only possible because of them.

I thank Cláudio Linhares, a great friend who was also a lab mate and collaborator during the past years. Let's continue this academic partnership! To Caio dos Santos, Everton Lira, Claudiney Tinoco, and Sara Melo, also great friends of mine and Ph.D. mates, my gratitude for the mutual help.

I thank CAPES for the financial support.

"Somewhere, something incredible is waiting to be known."
(Carl Sagan)

Resumo

Redes temporais (ou dinâmicas) são frequentemente usadas para modelar conexões que ocorrem ao longo do tempo entre partes de um sistema por meio de *nós* e *arestas*. Nessas redes, todos os nós, arestas e instantes de tempo são conhecidos e estão disponíveis para serem utilizados na análise. Entretanto, em várias situações reais, dados são produzidos de forma massiva e contínua, o que é conhecido como *fluxo contínuo de dados (FCD)*. Nesse tipo de aplicação, o volume de dados pode ser tão grande que o armazenamento deles pode ser impossível e as tarefas de mineração se tornam ainda mais desafiadoras. Em redes provenientes de FCD, arestas são continuamente adicionadas em distribuição não-estacionária. Tanto em redes temporais quanto em redes em FCD, padrões relacionados à atividade de nós e arestas são tipicamente irregulares ao longo do tempo, o que torna a *visualização* dessas redes útil para obter *insights* sobre a estrutura e dinâmica delas. Por outro lado, a distribuição não-estacionária aumenta a complexidade e torna a visualização de redes em FCD ainda mais desafiadora. Vários *layouts* visuais foram propostos até hoje, mas todos possuem limitações. O principal desafio é a quantidade de informação visual, que aumenta dependendo do tamanho e densidade da rede e causa poluição visual devido à sobreposição de arestas, resolução temporal e proximidade dos nós. Nesta tese, nós propomos métodos para melhorar a visualização de redes em FCD por meio da manipulação das três dimensões da rede: *nó*, *aresta* e *tempo*. Mais especificamente, nós propomos: (i) CNO, um método de ordenação de nós visualmente escalável; (ii) SEVis, um método de amostragem de arestas em FCD; (iii) um método para FCD que adapta a resolução temporal de acordo com níveis locais de atividade de nós. Também apresentamos um estudo comparativo considerando a combinação destes métodos. Por meio de estudos de caso com redes reais, mostramos que cada um dos métodos melhora bastante a legibilidade do *layout*, levando a uma tomada de decisão rápida e confiável.

Palavras-chave: Redes Temporais. Redes em Fluxo Contínuo. Visualização da Informação. Sumarização de Redes. Fluxo Contínuo de Dados. Comunidades.

Improving Visual Analysis of Streaming Networks

Jean Roberto Ponciano

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2020

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Ciência da Computação
Av. João Naves de Ávila, nº 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Tese de doutorado, 28/2020, PPGCO				
Data:	17 de setembro de 2020	Hora de início:	**13h30min**	Hora de encerramento:	**16h17min**
Matrícula do Discente:	11623CCP002				
Nome do Discente:	Jean Roberto Ponciano				
Título do Trabalho:	Improving Visual Analysis of Streaming Networks				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Ciência de Dados				
Projeto de Pesquisa de vinculação:	-				

Reuniu-se, por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Maria Camila Nardini Barioni - FACOM/UFU; Rodrigo Sanches Miani - FACOM/UFU; Zhao Liang - FFCLRP/USP; Luis Gustavo Nonato - ICMC/USP; Elaine Ribeiro de Faria Paiva - FACOM/UFU (coorientadora) e Bruno Augusto Nassif Travençolo - FACOM/UFU, orientador do candidato.

Os examinadores participaram desde as seguintes localidades: Zhao Liang - Ribeirão Preto/SP; Luis Gustavo Nonato - São Carlos/SP; Maria Camila Nardini Barioni - Glendale, Califórnia, Estados Unidos da América; Elaine Ribeiro de Faria Paiva e Bruno Augusto Nassif Travençolo - Uberlândia/MG. O discente participou da cidade de Uberlândia/MG.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Bruno Augusto Nassif Travençolo, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado

Esta defesa faz parte dos requisitos necessários à obtenção do título de Doutor.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Elaine Ribeiro de Faria Paiva**, **Professor(a) do Magistério Superior**, em 17/09/2020, às 16:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Luis Gustavo Nonato**, **Usuário Externo**, em 17/09/2020, às 16:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Zhao Liang**, **Usuário Externo**, em 17/09/2020, às 16:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Bruno Augusto Nassif Travençolo**, **Professor(a) do Magistério Superior**, em 17/09/2020, às 16:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Maria Camila Nardini Barioni**, **Professor(a) do Magistério Superior**, em 17/09/2020, às 16:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Rodrigo Sanches Miani**, **Professor(a) do Magistério Superior**, em 17/09/2020, às 16:26, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2264585** e o código CRC **F3DECB67**.

Abstract

Temporal networks (also known as *dynamic networks*) are often used to model connections that occur over time between parts of a system by using *nodes* and *edges*. In temporal networks, all nodes, edges, and times, are known and available to be used in the analysis. However, in several real-world applications, data are produced in a massive and continuous way, which is known as *data stream*. In this case, the volume of data may be so large that the storage may be impossible and mining tasks become more challenging. In *streaming temporal networks*, edges are continuously arriving in non-stationary distribution. In both temporal and streaming temporal networks, patterns related to node and edge activity are typically irregular in time, which makes the *visualization* of such networks helpful to gain insights about network structure and dynamics. Nevertheless, the non-stationary distribution of incoming data increases complexity and turns the streaming temporal network visualization even more challenging. Several visualization layouts have been proposed, but they all have limitations. The main challenge in this context is the amount of visual information, that increases depending on the network size and density, and causes visual clutter due to edge overlap, fine temporal resolution, and node proximity. In this thesis, we propose methods to enhance the visualization of streaming temporal networks through the manipulation of the three network dimensions, namely *node*, *edge*, and *time*. Specifically, we propose: (i) CNO, a visual scalable node ordering method; (ii) SEVis, a streaming edge sampling method; and (iii) a streaming method that adapts the temporal resolution according to local levels of node activity. We also present a comparative study considering the combination of these methods. We show through case studies with real-world networks that each of these methods greatly improves layout readability, thus leading to a fast and reliable decision making.

Keywords: Temporal Networks. Streaming Networks. Information Visualization. Network Sampling. Data Stream. Network Communities.

List of Figures

List of Tables

Acronyms list

AR Accept-Reject sampling

BVC Balanced Visual Complexity

CEO Chief Executive Officer

CNO Community-based Node Ordering

ComPAS Community Preserving sampling Algorithm for Streaming graphs

DyNetVis Dynamic Network Visualization System

EOD Edge Overlapping Degree

FF Fading Factor

KS Kolmogorov-Smirnov (statistic)

MSV Massive Sequence View

PHT-FM Page-Hinkley test with forgetting mechanism

PIES Partially Induced Edge Sampling

RFID Radio-Frequency Identification

RN Recurrent Neighbors

SEVis Streaming Edge Sampling for Network Visualization

SI Susceptible-Infected model

TAM Temporal Activity Map

Contents

Chapter 1

Introduction

Several researchers have been focused over the past years in problems related to data modeling and interaction with applications in distinct disciplines, such as computer science, biology, sociology, and others (ESTRADA, 2015). A popular approach for data representation is a network defined by nodes (instances) and edges (the relationship involving them) (ALBERT; BARABÁSI, 2002). In this way, a network may be used to represent the World Wide Web (web pages connected by hyperlinks), an organism cell (chemicals linked by chemical reactions), social interactions (any social relationship connecting individuals – e.g., friendship or collaboration), and many others (COSTA et al., 2011; ESTRADA, 2015).

In several situations, using only information about nodes and edges may not be enough to represent and comprehend the relations in the network. In social network analysis, for example, the information of *when* the connections occur may be crucial to describe such relations with less loss of context. Networks in which the time information is considered, in addition to nodes and edges information, are studied in several disciplines and received different nomenclatures: temporal networks, dynamic networks, time-varying networks (HOLME; SARAMÄKI, 2012). In this thesis, we will adopt the term *temporal network*.

In temporal networks, all nodes and edges are known and available to be used in the analysis (HOLME; SARAMÄKI, 2012). However, in several real-world applications, data are produced in a massive and continuous way. These types of data are referred as *data streams*. Examples include sensor monitoring, credit card transactions, phone calls, content sharing social networks, and others. In such applications, the volume of data may be so large that the storage may be impossible and mining tasks become more challenging (AGGARWAL, 2006). Temporal networks can be used to model such streaming data (AHMED; NEVILLE; KOMPELLA, 2013). Since edges are continuously arriving in non-stationary distribution and typically at high speed, such temporal networks are called *streaming temporal networks*, *streaming networks*, or *streaming graphs* (AHMED; NEVILLE; KOMPELLA, 2013; ZHANG, 2010; ESTRADA, 2013). The distribution of in-

coming data increases complexity and makes streaming network manipulation even more challenging.

Both temporal and streaming networks can be analyzed by adopting different strategies. Statistical analysis represents a common approach and is useful to identify specific trends and patterns in the data, being used, for example, in connection prediction (BU et al., 2019) and burst analysis (JO; HIRAOKA, 2019). When there is only a numeric output, however, it may represent a "black-box" to the user, thus impairing pattern comprehension. Another approach involves Information Visualization (CARD; MACKINLAY; SHNEIDERMAN, 1999; WARE, 2013), whose strategies try to assist the data analysis by providing interactive and graphical computational tools, thus including the user in the entire process of exploration and validation. An adequate Information Visualization strategy allows a visual analysis that is as much intuitive as possible and also helps the user in finding unexpected patterns, anomalies, and other behaviors in the data. Examples of information visualization methods applied to temporal and streaming networks can be found in (COL et al., 2018; NONATO; CARMO; SILVA, 2020; LINHARES et al., 2017b; ELZEN et al., 2014; GOROCHOWSKI; BERNARDO; GRIERSON, 2012; CRNOVRSANIN; CHU; MA, 2015; SARMENTO; CORDEIRO; GAMA, 2015a).

Visual analysis improves the understanding of the network dynamics and the identification of patterns and other properties, thus resulting in faster and more reliable decision making (ELZEN et al., 2013). Visualization strategies suitable for network analysis include the *Massive Sequence View (MSV)* layout (HOLTEN; CORNELISSEN; WIJK, 2007; ELZEN et al., 2013) and node-link diagrams (BATTISTA et al., 1994; CRNOVRSANIN; CHU; MA, 2015). Such strategies, that plot nodes and edges in the layout, suffer from visual clutter caused by overlapping edges, especially in large networks with hundreds or thousands of edges. Visual clutter refers to excessive items or information close together due to edge and node overlap (ELLIS; DIX, 2007). To reduce clutter in such layouts, some efforts focus on summarizing the network by changing its original temporal resolution (i.e., by grouping edges from subsequent timestamps – Figure 1(a)) (LINHARES et al., 2017b; ZHAO et al., 2018; LINHARES et al., 2019a; ROCHA; MASUDA; HOLME, 2017) or by selecting relevant edges for analysis (Figure 1(b)) (ZHAO et al., 2018; ROCHA; MASUDA; HOLME, 2017; ZHAO et al., 2019). Other strategies, on the other hand, focus on changing node positioning instead of summarizing the network. Their objective is to reduce visual clutter by reducing edge lengths (Figure 1(c)) (ELZEN et al., 2014; LINHARES et al., 2017b; GOROCHOWSKI; BERNARDO; GRIERSON, 2012; CRNOVRSANIN; CHU; MA, 2015).

MSV is a timeline-based layout in which the x-axis represents the timestamps and the y-axis represents the nodes of the network (Figure 1). Edges are represented by vertical lines between nodes over time. In this layout, nodes have fixed positioning over time (ELZEN et al., 2013). Such constraint is a problem for streaming scenarios, since

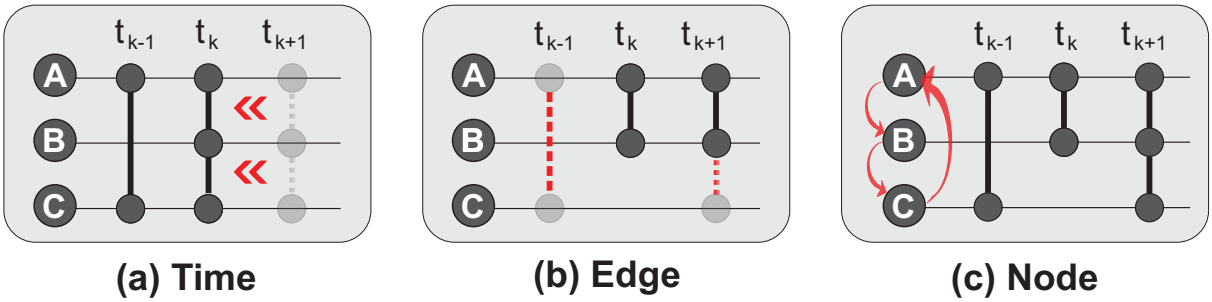

Figure 1 – Illustrative examples of the three network dimensions (namely *node*, *edge*, and *time*) manipulation over the MSV layout, in which the x-axis represents the timestamps and the y-axis represents the nodes of the network. In this layout, edges are represented by vertical lines between nodes over time. (a) Temporal resolution change. (b) Edge sampling. (c) Node ordering.

nodes would change positions in time as they gain or lose relevance to prevent overlapping edges and visual clutter. However, during streaming network analysis, relevant regions of interest can be perceived in the layout. Such regions can be treated as non-streaming temporal sub-networks and then can be further analyzed using any node ordering method. The node positioning affects the level of visual clutter in the layout and consequently the perception of patterns and decision making. Existing node ordering methods, however, are not visual scalable and so do not perform well when dealing with an elevated number of nodes and edges. Figure 2 shows the level of visual clutter caused by two different node ordering strategies for the same real-world network.

The temporal resolution plays an important role in the layout construction and, consequently, in the visual analysis. In several scenarios, as, for example, when the networks are temporally sparse, changing the resolution scale may facilitate the analysis and highlight patterns that would be difficult to see using the original resolution (LINHARES et al., 2017b; LEE; MOODY; MUCHA, 2019). The choice of the ideal temporal resolution, however, is not a trivial task. Since it is usually chosen before the construction of the layout, the user needs to be a domain specialist, in order to know *a priori* which resolution scale is adequate for the analysis given the expected edge distribution, otherwise it has to be empirically determined. In both cases, the resolution value is considered a global and static parameter that does not faithfully represent the variant level of activity (the number of edges and their distribution) over time. In streaming, the distribution of incoming edges is non-stationary, so the temporal resolution scale should adapt to the different levels of node activity. Even if we consider a global and static temporal resolution, exploratory analyses to assist the choice may not be possible when dealing with streaming networks because usually there are no *a priori* data to support the decision. Since the edge distribution changes over time, considering an initial set of nodes and edges of the stream to support the choice may be inefficient as well.

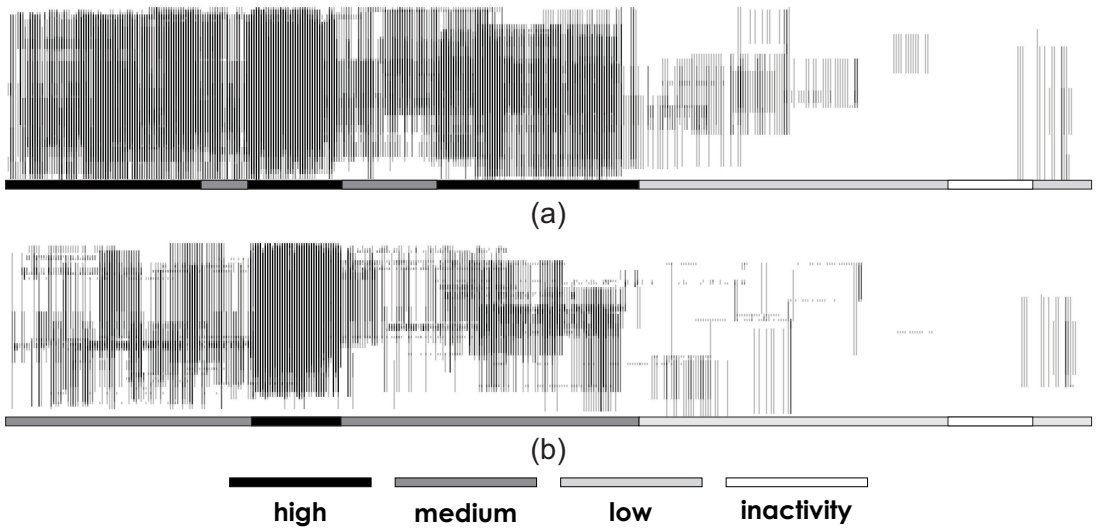

Figure 2 – MSV layouts generated by two different node ordering methods for the same real-world network. An adequate node positioning reduces visual clutter (removes noise) and helps to emphasise periods with different levels of activity over time. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, Temporal Network Theory, (LINHARES et al., 2019a) ©2019

Considering edge sampling, existing methods for temporal networks usually require all edges in primary memory (ZHAO et al., 2018; ZHAO et al., 2019) and may have computational time complexity $\mathcal{O}(m^2)$, where m is the number of edges (ZHAO et al., 2018; ZHAO et al., 2019). Such characteristics make their application in very large temporal or streaming networks infeasible. There are streaming edge sampling methods as well (AHMED; NEVILLE; KOMPELLA, 2013; ETEMADI; LU, 2019; AHMED et al., 2017). Some of them are focused in triangles estimation (e.g., (ETEMADI; LU, 2019)), outlier detection (e.g., (AGGARWAL; ZHAO; YU, 2011)), feature preservation (e.g., (SIKDAR et al., 2018)), and so on, but few approaches concern network visualization[1] (SARMENTO et al., 2016). Despite existing studies, network sampling remains an open research field (SARMENTO; CORDEIRO; GAMA, 2015b).

1.1 Motivation

Visualization techniques have been widely used in recent years to analyze temporal and streaming temporal networks that represent data from a variety of applications, such as corporate environments (ELZEN et al., 2013; ZHAO et al., 2018; SARMENTO; CORDEIRO; GAMA, 2015a), social media (CRNOVRSANIN; CHU; MA, 2015), and healthcare (LINHARES et al., 2017a).

[1] Streaming network visualization is also known as *online graph drawing* (SIMONETTO; ARCHAMBAULT; KOBOUROV, 2018).

Even with all edges and nodes available, improving the overall layout readability is already a challenging problem in temporal network visualization (ZHAO et al., 2018; ELZEN et al., 2013; LINHARES et al., 2017b). Most methods have high computational complexity or are unable to effectively reduce visual clutter while maintaining network properties. Such limitations become crucial when considering large networks in terms of the number of nodes and edges. In the case of streaming networks, this is even harder because edges are continuously arriving, typically at high speed, and in a way that the volume of network data does not fit in the primary memory (AHMED; NEVILLE; KOMPELLA, 2013). These characteristics increase complexity and bring new challenges to streaming network visualization, as, for example, the need for fast (often real-time) methods and the discard of an edge once it is processed (CRNOVRSANIN; CHU; MA, 2015).

There are methods in the literature that reduce visual clutter and improve layout readability by manipulating the previously mentioned network dimensions (*node*, *edge*, or *time*). However, many of them present limitations especially concerning their application in streaming scenarios due to their computational complexity. This thesis presents methods to enhance streaming network visualization by manipulating each of these network dimensions. We focus on streaming scenarios. However, since any method developed for streaming networks is also applicable in non-streaming scenarios (AHMED; NEVILLE; KOMPELLA, 2013), our methods can also be used when dealing with temporal networks with distinct characteristics such as small/large and sparse/dense network data.

1.2 Goals

Our main research goal is to develop methods to enhance the visualization of streaming networks. We decompose the main research goal in the following specific goals:

- Propose, implement and evaluate a streaming method that adapts the temporal resolution according to local levels of node activity over time (temporal dimension).
- Propose, implement and evaluate a visual scalable node ordering method, suitable for large networks (node dimension).
- Propose, implement and evaluate a streaming edge sampling method that discards less relevant edges while preserving the original edge distribution (edge dimension).

1.3 Hypothesis

We argue that it is possible to enhance streaming network visualization by reducing visual clutter. More specifically, we hypothesize that:

- Manipulating node, edge, or time network dimensions (Figure 1), or their combinations (Figure 3), enhance the visualization by reducing visual clutter.
- Using network community information (an important structural property of networks) as input of visualization methods reduces visual clutter.

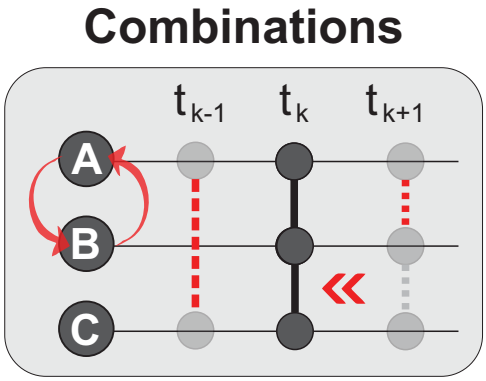

Figure 3 – Illustrative example of the combined network dimensions manipulation over the MSV layout. We hypothesize that it is possible to enhance the visualization by combining two or more dimensions manipulation (node ordering, time resolution change, and edge sampling).

1.4 Contributions

The contributions of this thesis are:

- A novel and adaptive temporal resolution method for streaming networks that considers local levels of node activity over time.
- A novel and visual scalable node ordering method, useful for the analysis of large networks.
- A novel edge sampling method, suitable for streaming networks, that discards less relevant edges while maintaining the characteristics of the original network in terms of edge frequency and distribution changes.
- A method for evaluating the performance of different network community detection algorithms through visual analysis.
- Presentation of case studies in which the combination of the manipulation proposals improved the visual analysis.
- A freely available extension of the software DyNetVis (LINHARES et al., 2017b) that incorporates our methods and also competing ones.

1.5 Outline

The chapters of this thesis are organized as follows.

Chapter 2. Presents concepts related to the three main research areas involved in our research: Networks, Information Visualization, and Data Streams. We present in details important definitions, such as *temporal* and *streaming temporal networks, visual clutter, network visualization, and network communities.*

Chapter 3. Discusses existing studies related to both temporal and streaming network visualization, specially those concerning network dimensions (*time*, *node*, and *edge*) manipulation.

Chapter 4. Presents our time manipulation method for streaming temporal networks. It is an automatic and adaptive temporal resolution method that changes the resolution based on the different levels of node activity over time.

Chapter 5. Presents our node manipulation method. It is a visual scalable node ordering method called *Community-based Node Ordering (CNO)*. CNO is useful for the analysis of large temporal networks. It combines network community detection with node ordering techniques to enhance the identification of visual patterns. As CNO uses community structure information, we initially present in this chapter a visual analysis method for evaluating the performance of different community detection algorithms.

Chapter 6. Presents our edge manipulation method for streaming temporal networks. It is a streaming edge sampling method for network analysis that discards less relevant edges while preserving the characteristics of the original distribution of edge counts. The method is flexible and can be used to improve a variety of layouts.

Chapter 7. Presents two case studies in which we combine our proposed methods to analyze real-world networks of different sizes. The combination considers methods for two and three network dimensions.

Chapter 8. Concludes the monograph, summarizing the main contributions and describing the author's bibliographical production.

Chapter 2

Background

In this thesis, we are interested in enhancing temporal and streaming network visualization by using methods that reduce visual clutter. To understand what this means, we first have to understand the main concepts from the three main related research areas (Figure 4): Networks, Information Visualization, and Data Streams. Our contributions lie at the junctions of these areas. Along this thesis, the terms "edge" and "connection" are used interchangeably. The same occurs with the terms "times" and "timestamps" and with "node ordering" and "node reordering".

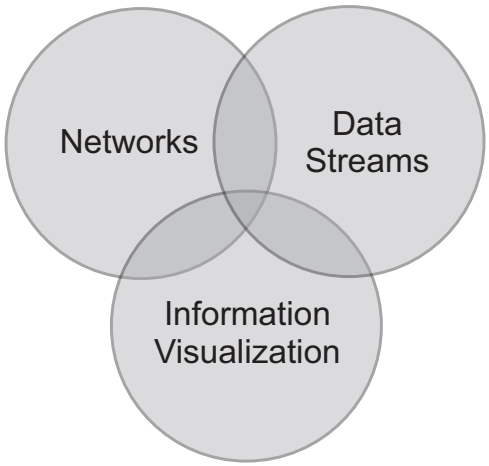

Figure 4 – Main research areas addressed in this thesis. Our contributions lie at their intersections.

2.1 Networks

Networks have been used to model a variety of systems in disciplines as diverse as computer science, biology, business, and sociology (ALBERT; BARABÁSI, 2002; ESTRADA, 2015; COSTA et al., 2011). A network is formed by nodes (representing the domain instances) connected by edges according to some rule. Examples of networks include the

Internet where computers are inter-linked by wires and routing devices, social networks where individuals are connected by social ties, flights between airports, or metabolic reactions in cells (ALBERT; BARABÁSI, 2002).

Mathematically, a network is represented by a graph $G = (V, E)$ as follows. $V = \{n_1, n_2, ..., n_N\}$ is the set of nodes in the network and $E = \{e_1, e_2, ..., e_M\}$ is the set of edges. Each edge $e_i = (n_x, n_y)$ connects two nodes $n_x, n_y \in V$ (COSTA et al., 2011). A network is a simplified representation of a system. Its nodes and edges can be labeled with additional information from metadata (e.g., node names or edge weights), but a lot of information is usually lost when reducing an entire system to a network (NEWMAN, 2010).

The structure of connections between nodes contains relevant information that helps to understand the function of the system (NEWMAN, 2010). Centrality measures, for instance, quantify the relevance of nodes or edges in the networked system (NEWMAN, 2010). Examples include *node degree*, defined by the number of edges incident (attached) to a node; *betweenness*, defined by the number of shortest paths (in terms of number of edges) passing through a node; and *closeness*, that measures how close a node is from the others (ESTRADA, 2015). Another important structural property of networks is the formation of communities. A network community is defined as a group of nodes that connect more often between themselves than with nodes from other groups (FORTUNATO, 2010). The next section provides more details about network communities.

2.1.1 Network Communities

The analysis of the network topology allows to extract relevant information, such as groups of elements and their hierarchical organization (FORTUNATO, 2010; CAZABET; ROSSETTI, 2019). Groups whose members interact more between themselves than with others in the network are known as communities. As example, communities may correspond to classmates, age-groups, proteins with similar functions, brain regions, related diseases, among others (FORTUNATO; HRIC, 2016). Network community detection can be seen as a clustering task, highly used in data mining scenarios, but applied to networks (GUIDOTTI; COSCIA, 2017). In this way, traditional clustering methods, such as *Density-based Spatial Clustering of Applications with Noise* (DBSCAN) (ESTER et al., 1996), can be adapted to the community detection task (GIALAMPOUKIDIS et al., 2016). Figure 5 shows a network with 13 nodes and 3 communities (each one represented by blue, green or red nodes). The network was drawn using a node-link diagram, where circles linked to each other by straight lines represent nodes and the connections (edges) involving them (BATTISTA et al., 1994).

The identification of network communities is an important but complex task (FORTUNATO; HRIC, 2016). There are several proposals in the literature related to community detection. In (FORTUNATO; HRIC, 2016), the authors evaluated several com-

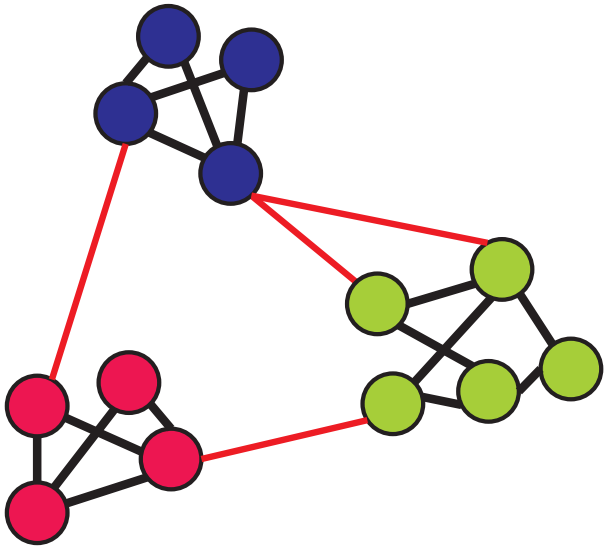

Figure 5 – Network represented by a node-link diagram with three communities (each one represented by blue, green or red nodes) and 13 nodes. Red edges connect nodes from different communities. A network community is defined as a group of nodes that interact more often between themselves than with nodes from other groups.

munity detection methods with different detection procedures. Two of the most recommended techniques in their study are Infomap (ROSVALL; BERGSTROM, 2008) and Louvain (BLONDEL et al., 2008), methods also recommended by other studies (MOTHE; MKHITARYAN; HAROUTUNIAN, 2017; ORMAN; CHERIFI; LABATUT, 2011; ORMAN; LABATUT; CHERIFI, 2012). These recommendations are based mostly on the quality of the results and response time. In fact, both methods have computational time complexity $\mathcal{O}(m)$, where m is the number of edges, which allows their application in large networks (FORTUNATO, 2010). Besides, there are freely available implementations of them for different platforms in the Internet.

Infomap and Louvain employ different detection procedures. Therefore, they are categorized in different classes of algorithms (ROSVALL et al., 2019; DRIF; BOUKERRAM, 2014; ORMAN; CHERIFI; LABATUT, 2011; ORMAN; LABATUT; CHERIFI, 2012). According to Rosvall et al. (2019), Infomap is categorized as a dynamical perspective method. Methods in this category use dynamical processes (e.g., random walk) and identify coarse-grained descriptions in the network. In another categorization, Infomap is considered as a stochastic method based on a dynamic process (DRIF; BOUKERRAM, 2014). The network structure is represented by a Huffman code, so the process of finding communities is based on minimizing the amount of required information to represent a random walk in the network. The Louvain method, on the other hand, is deterministic and based on modularity optimization (DRIF; BOUKERRAM, 2014). Moreover, as it is based on clustering, it maximizes internal density, producing groups of densely connected nodes (ROSVALL et al., 2019). Louvain is a greedy, hierarchical and agglomerative ap-

proach of community detection based on the modularity measure. Initially, each node is considered as a community. From there, for each node, it is computed the modularity gain of moving it to neighbor communities. This displacement process repeats until no node is moved. In this step, the Louvain algorithm considers each detected community as a node. The process repeats in this new induced network until no gain is detected and no node changes its position.

Network communities represent valuable information related to network structure. Such information is already used, for instance, in network visualization (see Section 2.2.1) (PORTER; ONNELA; MUCHA, 2009; AHN; BAGROW; LEHMANN, 2010; BASAILLE et al., 2016) and in community evolution analysis (as in the case of temporal networks – see next section) (MITRA; TABOURIER; ROTH, 2012; ORMAN et al., 2014; ROSVALL; BERGSTROM, 2010; SALLABERRY; MUELDER; MA, 2013).

2.1.2 Temporal Networks

In some cases, the static structure is not sufficient to fully capture the complexity of interactions between nodes. For example, in networks that evolve in time, the timings of events or the times in which edges are active also provide valuable information about the dynamic evolution of the system. Typical temporal patterns include, for example, burst interactions and circadian rhythms in social networks, or birth and death of nodes and edges observed in most real-world networks (HOLME; SARAMÄKI, 2012). Such dynamic networks are also called time-varying or temporal networks.

A temporal network is represented by $G = (V, E)$, where $V = \{n_1, n_2, ..., n_N\}$ is the set of nodes in the network and $E = \{e_1, e_2, ..., e_M\}$ is the set of edges. Each edge $e_i = (n_x, n_y, t_k)$ connects two nodes $n_x, n_y \in V$ at a particular time t_k (LINHARES et al., 2019a; HOLME; SARAMÄKI, 2012). Considering t_{final} as the end of the observation period, $0 \leq t_k \leq t_{final}$. The connection time t_k is discrete, and so an edge that occurs at t_k actually occurs in the interval $[t_k, t_k + \delta)$, where δ is the temporal resolution (LINHARES et al., 2019a). In another definition of temporal networks, each of the edges is represented by $e_i = (n_x, n_y, t_{ini}, t_{end})$ (HOLME; SARAMÄKI, 2012). In this case, edges are not active over a set of times, but over a set of intervals (PEREIRA et al., 2018). Such temporal network is known as *interval graph* (HOLME; SARAMÄKI, 2012). In this thesis, we rely on the first definition, in which an edge occurs in a particular timestamp, as some softwares and several freely available networks consider this format (real-world networks that will be explored along this thesis are presented below). To simplify, self-edges (i.e, edges connecting a node to itself) are removed (LINHARES et al., 2019a). In the same way, multiple edges (i.e., two or more edges that connect the same two nodes at the same time) could be summed and assigned as weight to a single edge.

2.1.3 Examples of Real-world Temporal Networks

One can find several temporal networks in the literature. Along this thesis, we consider networks of distinct sizes and densities, and from a variety of domains (e.g., face-to-face interactions in hospital and school environments, email exchange in a corporate company, and social networks).

In this thesis, the data of the networks that contain face-to-face interactions were collected via *Radio-Frequency Identification (RFID)* sensors. When two individuals have a frontal approximation from 1 up to 1.5 meters with the RFID sensors, one contact between them is registered in a time interval of 20 seconds. In the same way, when there is no trade information between the RFID sensors, the contact between these individuals is interrupted (CATTUTO et al., 2010; VANHEMS et al., 2013). Therefore, in this type of network, individuals are represented by nodes and the connections among them are represented by edges.

Primary School

The *Primary School* network (GEMMETTO; BARRAT; CATTUTO, 2014; STEHLÉ et al., 2011) represents contacts involving teachers and students of a primary school located in Lyon, France, between October 1^{st} - 2^{nd} of 2009. This network contains 242 nodes and 125,773 edges distributed in 5,846 timestamps. The original temporal resolution is 20 seconds, which means that each timestamp in resolution 1 comprises a 20 seconds interval. The data represent contacts from the first to fifth grade, each of them having two classes (A and B), totalizing 10 classes. In this network, the majority of edges occurs among students of the same class and each class has an assigned teacher (STEHLÉ et al., 2011).

Enron

The *Enron* network (KEILA; SKILLICORN, 2005; SHETTY; ADIBI, 2004) contains email communications from Enron Inc., a former energy company involved in the biggest American accounting fraud (SUN et al., 2007). The network is composed by 148 nodes (Enron employees) and 24,667 edges (email exchanges between employees) distributed in 1,346 timestamps (each timestamp refers to a 1-day interval in the original temporal resolution).

Hospital

The *Hospital* network (VANHEMS et al., 2013) was collected in a geriatric unit of a university hospital, in Lyon, France, between Dec 6, 2010 (Monday) and Dec 10, 2010 (Friday). The data include face-to-face approximations of 46 employees and 29 patients,

organized in profiles: 11 employees are medical doctors (physicians or interns - MED profile), 27 are nurses or nurses' aides (NUR profile) and 8 are administrative staff (ADM profile). The group of patients was associated to the PAT profile. In total, the network has 75 nodes and 32,424 edges. The original temporal resolution of the network is 20 seconds, i.e., each timestamp in the network refers to a 20-second interval and all edges between two nodes that occur in this interval are considered as one.

Twitter

The *Twitter* network is a subset of the network presented in (PEREIRA; AMO; GAMA, 2016a) that contains data from retweets that mention a Brazilian newspaper called *Folha de São Paulo*. When a user retweets someone's tweet related to news published by the newspaper, an edge is created in the respective timestamp. Such edge connects the two users involved in the retweet process, so these users become nodes in the network. The network is composed by 50,514 nodes and 107,948 edges collected between July 12, 2016 and July 21, 2016. All edges are classified in topics related to the content of the tweet that each of them represents (Sports, Celebrity, Corruption, Politics, Education, Security, or International). The original temporal resolution of the network is 1 hour, i.e., each timestamp in the network refers to a 1-hour interval and all edges between two nodes that occur in this interval are considered as one.

Museum

The *Museum* network (ISELLA et al., 2011) is composed of data related to face-to-face approximations among people visiting the *Science Gallery* in Dublin, Ireland. The network contains 72 nodes and 6,980 edges distributed in 1,312 timestamps. Each timestamp in this network refers to a 20-second interval in the original temporal resolution (Res. 1).

Sexual

The *Sexual* network (ROCHA; LILJEROS; HOLME, 2010; ROCHA; LILJEROS; HOLME, 2011) is composed of data related to sexual encounters between heterosexual males and female prostitutes. The data was collected through posts in a public and online forum. Each post comments/evaluates the encounter, and so each one comprehends an edge that connects the involved individuals (nodes) in the network. There are 12,157 nodes and 34,060 edges distributed in 1,000 timestamps.

2.1.4 Reducing Temporal Network Data

Many factors make infeasible the analysis of entire temporal networks (AHMED; NEVILLE; KOMPELLA, 2013; ROCHA; MASUDA; HOLME, 2017). First, real-world temporal networks may be massive in size (in terms of number of nodes $|V|$ and edges $|E|$) and thus the memory cost may become problematic (ROCHA; MASUDA; HOLME, 2017). Second, there are networks that can only be accessed through crawling (e.g., World Wide Web) and others that are not completely visible (e.g., Facebook data) (AHMED; NEVILLE; KOMPELLA, 2013). Moreover, sampled networks may be used in computationally intensive simulations or lead to better visual analyses (ROCHA; MASUDA; HOLME, 2017; NEWMAN, 2010).

Three aspects may be considered when sampling temporal networks (ROCHA; MASUDA; HOLME, 2017): (i) which nodes and/or which edges of the original network will be considered; (ii) the observation time that will be adopted in the network analysis (e.g., 1 hour or 1 day of the network); (iii) in which temporal resolution the data will be recorded. Figure 6 shows a timeline representation of a temporal network to illustrate the three mentioned aspects that may be considered when sampling a temporal network.

Changing the temporal resolution means that edges from subsequent timestamps will be grouped in a single time (ROCHA; MASUDA; HOLME, 2017). Linhares et al. (2017b) present a manner of changing the temporal resolution scale of a network by computing a new time for each edge (Equation 1):

$$t_{\text{new}} = \left\lfloor \frac{t_{\text{ori}} - t_{\text{s}}}{\delta} \right\rfloor \delta + t_{\text{s}} \tag{1}$$

where t_{new} is the new timestamp of the edge, t_{ori} is the timestamp of the edge from the original temporal resolution, t_{s} is the first timestamp of the network and δ is the desired scale of time (resolution factor). When changing the temporal resolution, each new timestamp must represent a time interval that is a multiple value of the original resolution. Moreover, repeated edges are considered as a single one if their timestamps are merged. By adopting a different resolution scale, one may identify temporal patterns that would be difficult to see in the original resolution, especially in temporally sparse networks (LINHARES et al., 2017b). The resolution change process is exemplified in Figure 6(d), where in resolution 2 (new resolution defined by $\delta = 2$) each pair of adjacent timestamps are merged into one, thus the edges (A,B,0) and (A,B,1) become a single connection ((A,B,0) resolution 2) and so on.

2.2 Information Visualization

A picture is worth a thousand words. A single picture contains information that is processed faster than a comparable page of words. While the human perceptual system

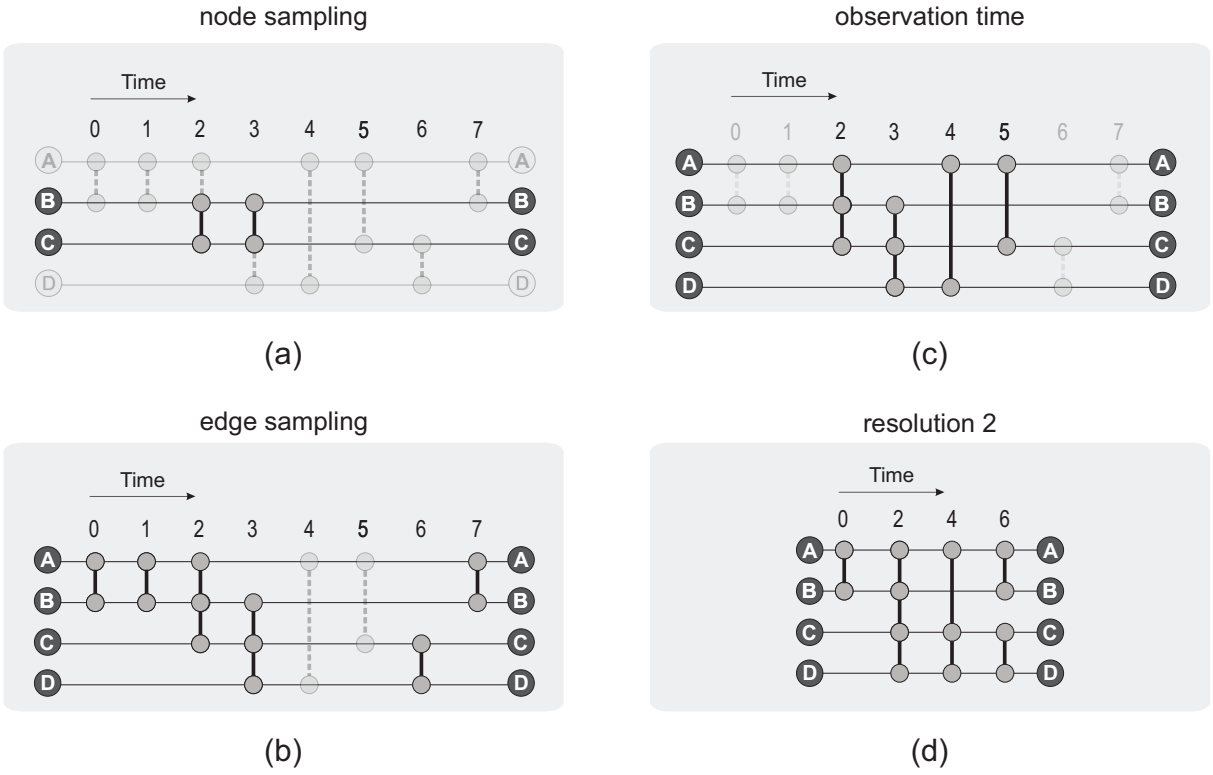

Figure 6 – Illustration of temporal network sampling using a timeline-based representation (horizontal lines represent nodes and straight vertical lines represent edges connecting two nodes). (a) Node sampling – only those edges connecting nodes B and C are considered. (b) Edge sampling – only those edges that connect nodes positioned close to each other in the screen are considered. (c) Choosing an observation time – only those edges from time 2 to time 5 are considered. (d) Temporal resolution change from $\delta = 1$ to $\delta = 2$ – repeated edges are considered as a single one if their timestamps are merged.

interprets an image in parallel, the text analysis depends on the sequential process of reading (WARD; GRINSTEIN; KEIM, 2015). In this context, *visualization* can then be defined as the use of graphical representations to transmit information (WARD; GRINSTEIN; KEIM, 2015).

The visualization research field comprehends two main sub-fields (WARD; GRINSTEIN; KEIM, 2015): scientific visualization and information visualization. *Scientific visualization* is applied to scientific data and usually is physically based (CARD; MACKINLAY; SHNEIDERMAN, 2009). It can be used to view and analyze data in areas as diverse as fluid dynamics, molecular modeling, and geophysics (UPSON et al., 1989). On the other hand, *information visualization* aims to bring meaningful visual representations from abstract and non-spatial data (e.g., time-series data, networks, and trees). Generating meaningful and spatial graphical representations from such abstract data is the major challenge in the information visualization field (CARD; MACKINLAY; SHNEIDERMAN, 2009; CHEN, 2010).

Information visualization can be defined as *"the use of computer-supported, interactive, visual representations of abstract data in order to amplify cognition"* (CARD; MACKINLAY; SHNEIDERMAN, 1999). Its main purpose is to allow users to gain insights – which includes finding unexpected discoveries and a deepened understanding (CHEN, 2010) –, draw conclusions, and interact with the data (KEIM, 2002).

Information visualization involves the design, the development, and the application of such visual representations (CHEN, 2010). Figure 7 presents the visualization process proposed in (CARD; MACKINLAY; SHNEIDERMAN, 1999). First, the raw data is processed and transformed in filtered and normalized data tables that contain a unified structure. Then, visual mappings are performed to represent data using graphical entities. These visual structures are then transformed (by changing position, rotation, scaling, etc) to provide a set of views that can be explored by the user. Data transformations, visual mappings and view transformations are interactive processes, so the user can adjust each of them according to his/her tasks and needs.

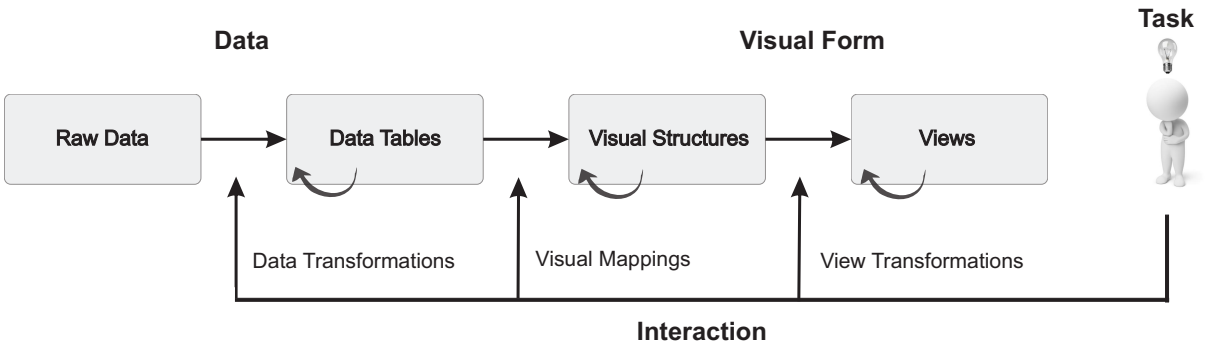

Figure 7 – Visualization process – reference model for visualization proposed in (CARD; MACKINLAY; SHNEIDERMAN, 1999).

According to Keim (2002), visual data exploration usually provides faster data exploration and better results when compared to automatic data mining techniques. The author explains that this happens because visual data exploration can easily deal with inhomogeneous and noise data, is intuitive and requires no understanding of complex mathematical or statistical algorithms. Moreover, the degree of confidence in the findings is higher when adopting visual representations. In this thesis, we are particularly interested in the visual exploration of temporal networks, subject that will be further explored in the following.

2.2.1 Temporal Network Visualization

The employment of an effective temporal network visualization strategy helps the user in the network evolution comprehension. Its main purpose is to facilitate the identification of patterns, anomalies, and other network properties, thus enhancing decision making.

In this context, several visual strategies may be adopted (BECK et al., 2017), such as matrix-based (BACH, 2016; BEHRISCH et al., 2016) and circular approaches (ELZEN et al., 2014), node-link diagrams (BATTISTA et al., 1994; LINHARES et al., 2017b) and Massive Sequence View (MSV) layouts (LINHARES et al., 2017b; ELZEN et al., 2013). Among these proposals, node-link diagrams and MSV are the most used strategies to analyze edge distribution over time (LINHARES et al., 2019a).

The node-link diagram (also known as *Structural Layout* (LINHARES et al., 2017b; LINHARES et al., 2019a)) is the conventional network representation in which the nodes are spatially placed on the layout with edges connecting them. Edges may contain a list of numbers representing the times they are active (LEE; MOODY; MUCHA, 2019) (see Figure 8(b)). Using node-link diagram is recommended to get an overview of the network since it facilitates the identification of multiple structures at the same time. In this layout, it is recommended to use nodes with fixed positions. Figure 8(c) shows the network evolution using small multiples (BACH; PIETRIGA; FEKETE, 2014) over the node-link diagram. In practice, the user may also analyze the network evolution using animation (BECK et al., 2017). The animation frames are updated at each time, i.e., for each time, only the nodes and edges that are active at that moment are highlighted in the layout. Nodes and edges from the past time(s) may also be shown with opacity to facilitate comprehension of the changes in the network structure.

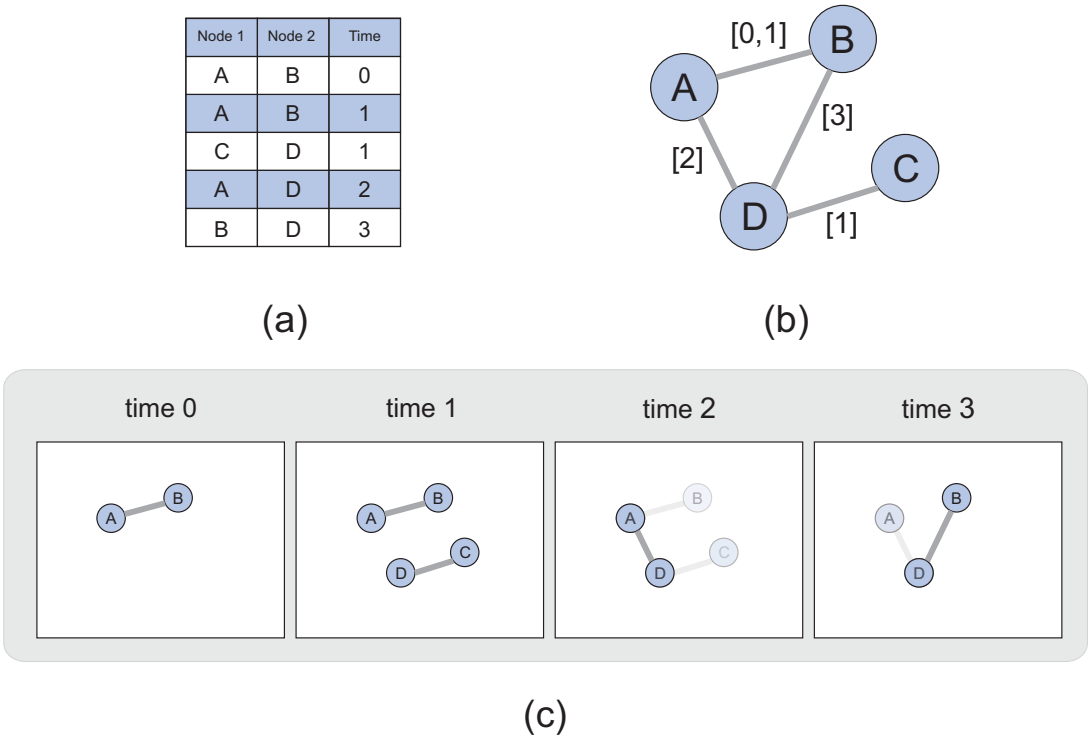

Figure 8 – Network evolution representation using node-link diagram. (a) Tabular (raw) data. (b) Node-link diagram with edges containing a list of numbers representing the times they are active. (c) Small multiples over the node-link diagram.

The Massive Sequence View (MSV), also known as *Temporal Layout* (LINHARES et al., 2017b; LINHARES et al., 2019a), is a timeline-based layout (BECK et al., 2017) in which the x-axis represents the timestamps and the y-axis represents the nodes of the network. In this layout, nodes cannot change their positions over time. Every time there is a connection (edge) between a pair of nodes, a vertical line is drawn linking them in the respective timestamp. The construction of the MSV layout from the tabular (raw) data is illustrated in Figure 9(a,b).

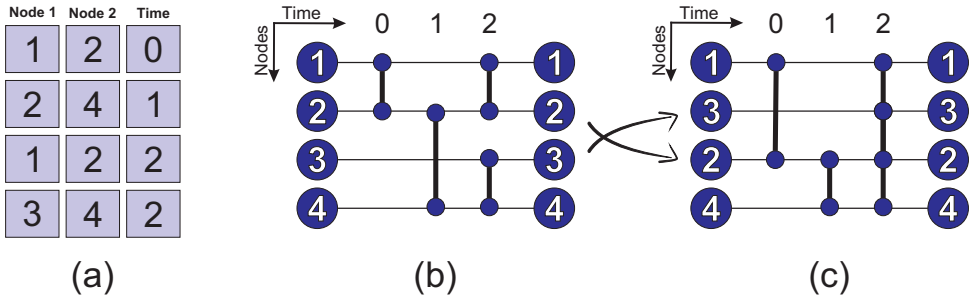

Figure 9 – Network representation using the MSV layout. (a) Tabular data. (b) MSV layout, in which the x-axis represents the time and the y-axis represents the nodes, respectively. (c) The same visual representation of (b) but with a different node ordering. As it can be seen, the node ordering influences the generated layout and, consequently, the identification of patterns. Reprinted from (PONCIANO et al., 2020) ©2020 Vilnius University Institute of Data Science and Digital Technologies.

A relevant aspect that should be considered in network visualization is the user's mental map preservation, i.e., the differences between consecutive timestamps (or frames, if animation) must be minimal (LIN; LEE; YEN, 2011). In other words, the node positioning (and consequently the edge positioning) should be as preserved as possible (SAFFREY; PURCHASE, 2008) to provide a faster comprehension of the changes in the network structure. MSV preserves the user's mental map better than animated node-link diagrams (LINHARES et al., 2019a).

An important property of the MSV layout is the node positioning. By comparing the layouts from Figure 9(b-c), one can see changes in the edge lengths due to differences in the node positioning. In real-world networks, node ordering highly affects the perception of patterns (Figure 10). In the following we discuss about visual clutter and show how it impairs pattern identification.

2.2.2 Visual Clutter in Temporal Network Visualization

In the context of networks, visual clutter refers to excessive items or information close together due to edge and node overlap (ELLIS; DIX, 2007). When applied to real-world networks with a large amount of data, MSV suffers from visual clutter caused by overlapping edges, and thus important patterns may not be perceived (Figure 10). The

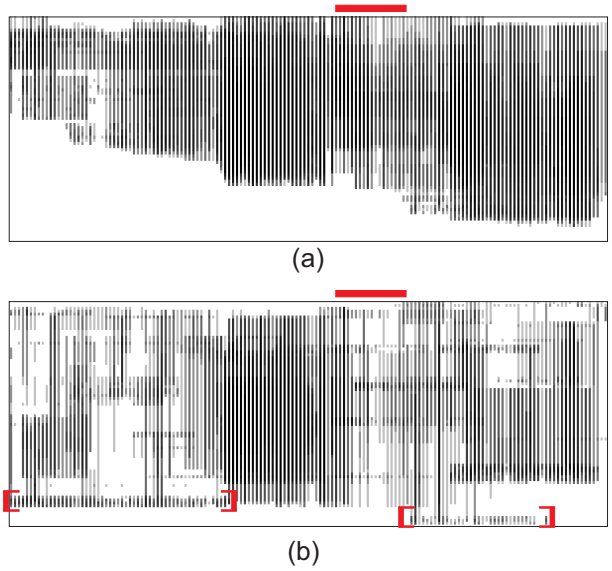

Figure 10 – Different levels of visual clutter caused by different node ordering methods for the same real-world network. In the time interval indicated by the red bar, one may see that the layout from (a) presents a higher level of visual clutter when compared with (b). Moreover, the layout from (b) allows the identification of groups of nodes (red brackets) that are not identified in (a).

Temporal Activity Map (TAM) (LINHARES et al., 2017b), which omits all edges and changes the shape of nodes from circles to squares, represents an alternative layout useful to identify patterns based only in the node activity, ignoring which nodes are involved in the connections (Figure 11(b)). However, when the information of connections are still needed or the amount of information is still large, these issues must be solved with strategies that reduce the amount of visual information by sampling the network or by reordering nodes.

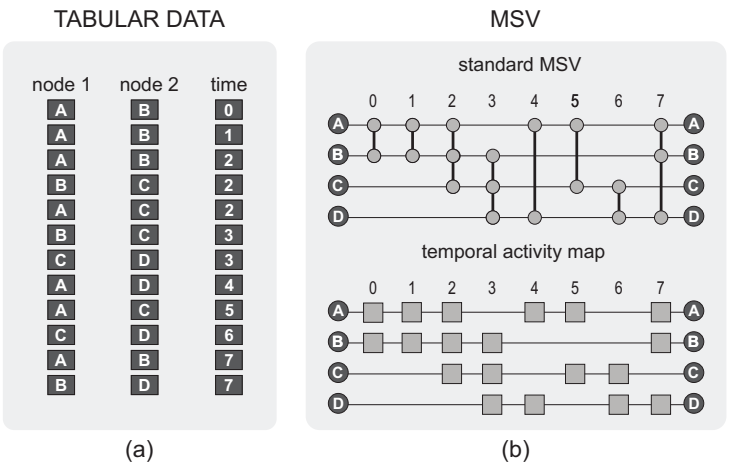

Figure 11 – Network representation using MSV and TAM layouts. (a) Tabular (raw) data; (b) MSV layout: standard MSV, showing nodes and edges, and temporal activity map (TAM) with all edges removed.

Node-link diagrams are also affected by visual clutter. Figure 12 shows two node-link diagrams with different levels of visual clutter. To reduce clutter in node-link diagrams, a number of strategies can be employed, from methods to change nodes' positions, such as force-based and circular algorithms (SIX; TOLLIS, 2006; BATTISTA et al., 1994; MI et al., 2016), to hierarchical simplification (DIAS et al., 2017) and edge-bundling strategies (HOLTEN; WIJK, 2009; LAMBERT; BOURQUI; AUBER, 2010; LHUILLIER; HURTER; TELEA, 2017).

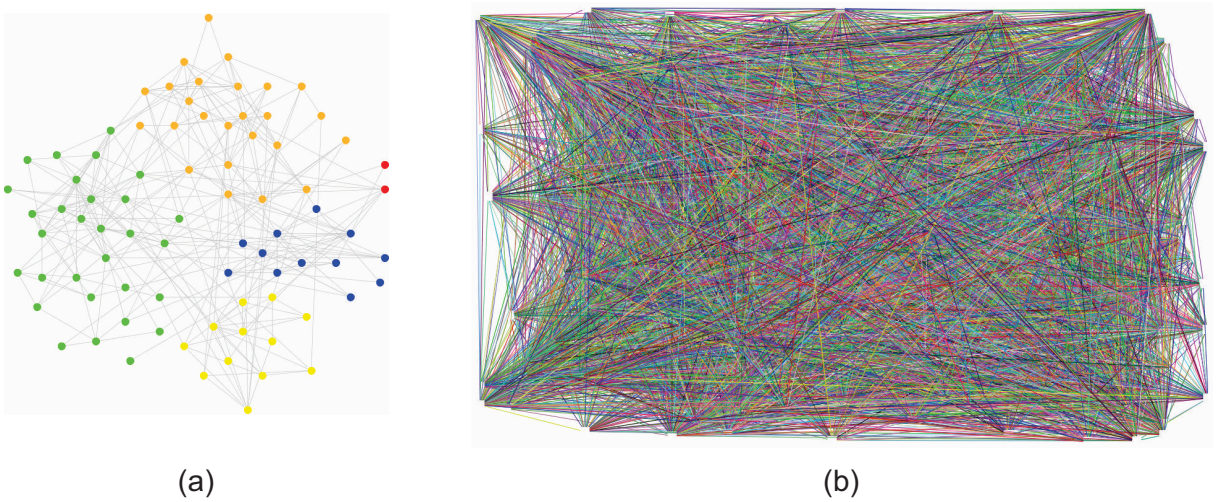

Figure 12 – Two node-link diagrams with different levels of visual clutter. The diagrams correspond to (a) a node-link diagram where some information about the structure can be retrieved and (b) a node-link diagram where no meaningful information can be retrieved. Reprinted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature, Temporal Network Theory, (LINHARES et al., 2019a) ©2019.

Less visual information usually leads to less visual clutter, and so strategies such as node ordering, edge sampling, and temporal resolution change can benefit visual exploration. Figure 13 shows the effect of edge sampling in both node-link diagram and MSV layout. Figure 13(b) shows the network evolution using small multiples over a node-link diagram. Figure 13(c) illustrates an edge sampling process using the node-link diagram. At time 1, the edge (B,C) is removed, but the three existent nodes (A, B, and C) remain active since there are incident edges to all of them. At time 2, with the removal of the edge (B,A), node B has no incident edges and is thus discarded. At time k, after (A,E) and (D,C) removal, the four active nodes are maintained in the layout. Figure 13(d) illustrates the MSV layout for the same network. In the figure, edges are positioned side-by-side for a comparative analysis. As already mentioned, in practical applications using MSV layouts, the edges overlap each other generating visual clutter. Figure 13(e) illustrates the same edge sampling used in Figure 13(c), but now applied to MSV. With fewer edges the layout becomes cleaner, facilitating the identification of patterns.

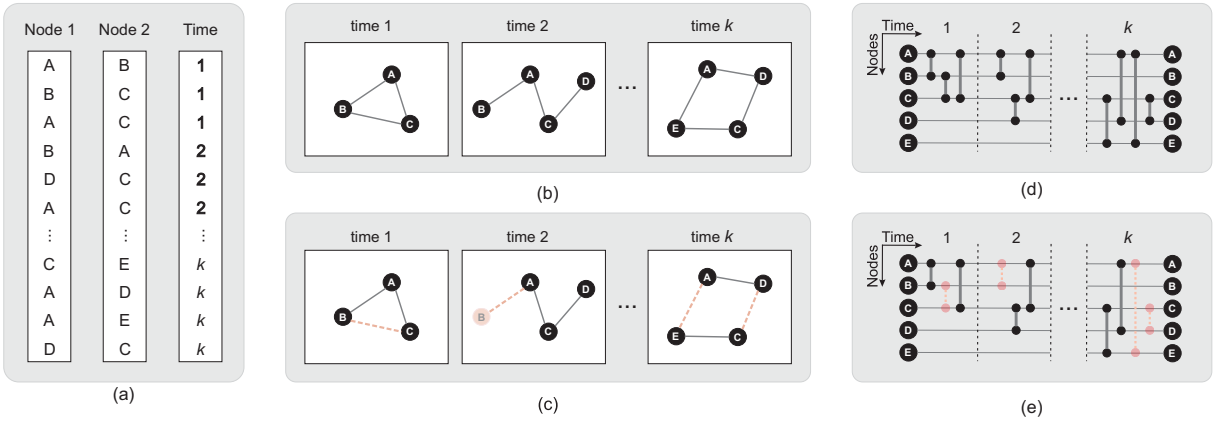

Figure 13 – Temporal network representation using both node-link diagram and MSV layouts. (a) Tabular data. The node-link diagram with (b) the original edges and (c) sampled edges. Dashed lines represent removed edges and pink colored nodes represent removed nodes. The MSV layout with (d) the original edges and (e) sampled edges. Pink edges and nodes represent removed elements.

2.3 Data Streams

Several real-world applications produce massive and continuous data. Examples include sensor monitoring, credit card transactions, phone calls, content sharing social networks, and many others. In such cases, the volume of data is so large that the storage may be impossible and mining tasks become more challenging (AGGARWAL, 2006). These types of data are referred as *data streams* (AGGARWAL, 2006).

In data streams (BABCOCK et al., 2002; HAN; KAMBER, 2006):

- The elements arrive online and usually at high speed, flowing in and out in a fixed order;
- The system has no control over the order in which the elements arrive to be processed;
- The size (of a data stream) is potentially unbounded;
- The element of a data stream is discarded or archived after being processed. An element cannot be retrieved easily unless it is explicitly stored in memory, whose size is smaller than the size of the stream.
- Only one or a small number of scans in the data is allowed in streaming algorithms. Fast (often real-time) response is required.

Furthermore, in streaming scenarios, data are continuously generated by dynamic environments and in non-stationary distributions (GAMA, 2010). As a consequence,

new classes/groups of elements may appear or disappear over time as well as known classes/groups may evolve.

Formally, a data stream is composed of a set of multi-dimensional elements $X_1, ..., X_k, ...$ arriving at timestamps $t_1, ..., t_k, ...$ Each element contains d dimensions and is denoted by $X_i = (x_i^1, ..., x_i^d)$ (AGGARWAL et al., 2003). In the streaming model, data arrive sequentially and thus create an ordered sequence of events. Such sequence may or not be related to a concrete notion of time (PAIVA, 2014). Customer shopping sequences, for instance, represent ordered sequences of events, but, since adjacent observations are not dependent, the concrete notion of time may be not present.

Since the massive volume of elements in a stream does not fit in primary memory, one possibility is to manipulate short windows of elements. There are two basic types of windows (AGGARWAL, 2006; GAMA, 2010):

- Landmark window: composed of every element from a starting time i to the current time t (window $W[i, t]$). If i refers to the first time of the stream, then the window comprehends the entire data stream (AGGARWAL, 2006). This type of window, used in studies such as (MANKU; MOTWANI, 2002; LIU; GUAN; HU, 2009; ACKERMANN et al., 2012), treats each timestamp after i equally important (AGGARWAL, 2006).

- Sliding window: moves along with the current time point t. This type of window contains only the most recent elements and is defined by $W[t - w + 1, t]$, where w is the window size. This type of window, used in studies such as (REN; MA, 2009; AMINI; WAH; TEH, 2012; BRAVERMAN; OSTROVSKY; ZANIOLO, 2009), gives more importance to recent elements (elements before time $t - w + 1$ are not of interest) (AGGARWAL, 2006).

There are several studies that handle data streams with focus on clustering tasks (e.g., (KRANEN et al., 2011; ACKERMANN et al., 2012; CAO et al., 2006)), management and querying (e.g., (BABU; WIDOM, 2001; GREENWALD; KHANNA et al., 2001)), novelty detection (e.g., (MASUD et al., 2011; AL-KHATEEB et al., 2012)), visualization (e.g., (KRSTAJIĆ; KEIM, 2013; MANSMANN et al., 2012)), and others. In this thesis, we are particularly interested in manipulating streaming temporal networks, which are composed of streaming data that represent interactions among elements. More details will be presented in the next section.

2.3.1 Streaming Temporal Networks

Streaming data that represent interactions among elements may be naturally represented as a *streaming temporal network* (AHMED; NEVILLE; KOMPELLA, 2013; MCGREGOR, 2009). In a telecommunication context, phone calls forms a network between

the involved phone numbers (ZHANG, 2010). In the same way, web pages and the links between them form a web network. For convenience, we will adopt the term *streaming network* instead of *streaming temporal network* from now on.

In non-streaming temporal networks (or simply temporal networks), all edges and nodes are known and available to be used in the analysis (see Section 2.1.2). A temporal network has a delimited observation period, its data usually fits in primary memory and unrestricted random access is allowed. In streaming, the distribution of incoming edges is non-stationary and changes over time (AHMED; NEVILLE; KOMPELLA, 2013). Edges are continuously arriving, typically at high speed, and in a way that the volume of data does not fit in the primary memory. Formally, we define a streaming network $S = \langle e_1, e_2, ..., e_m, ... \rangle$ as an undirected and unweighted temporal network $G = (V, E)$ as follows: $e_i = (n_x, n_y, t_k)$, $e_i \in E$, represents an edge that occurs at time t_k, $0 \leq k \leq \infty$, between the pair of nodes (n_x, n_y), where $n_x, n_y \in V$ and $|V| \rightarrow \infty$. Note that it is possible to have more than one edge per time. This definition is different from the ones that consider each edge arriving in a different time (CRNOVRSANIN; CHU; MA, 2015; AHMED et al., 2017; ETEMADI; LU, 2019).

Only sequential access in the stream is possible. In the workspace (portion of the stream in the primary memory), however, the streaming algorithm can perform unrestricted random access (ZHANG, 2010). Methods for processing streaming networks require efficient and real-time processing. This means that a streaming algorithm, besides the restricted access to the stream data (edges), must process the stream in a single scan, or in a small number of scans (ZHANG, 2010).

2.4 Final Considerations

This chapter presented concepts related to the three main research areas involved in our research: Networks, Information Visualization, and Data Streams. Now, one can fully understand our proposal: *to enhance streaming network visualization by using methods that reduce visual clutter*. We present in the next chapter the state-of-the-art related to temporal and streaming network visualization, more specifically those studies concerning network dimensions (*node*, *time*, and *edge*) manipulation, which includes node ordering, temporal resolution change, and edge sampling strategies.

Chapter 3

Related Work

Our research aims to enhance network visualization by manipulating *node*, *time*, and *edge* dimensions. In this sense, this chapter presents studies that propose strategies that manipulate these dimensions in temporal networks (Section 3.1) and in streaming networks (Section 3.2). We also present DyNetVis (LINHARES et al., 2017b), an interactive software for visualizing temporal networks (Section 3.3).

3.1 Manipulating Dimensions in Temporal Networks

Although both node-link diagram and *Massive Sequence View (MSV)* are suitable for exploring temporal networks, MSV (Figure 14(b)) is preferred because it better maintains the user's mental map of the network evolution in comparison to the node-link diagram, even when using animation (LINHARES et al., 2019a). As discussed in Section 2.2.2, the MSV layout suffers from a high level of visual clutter when applied to real-world applications. Therefore, all three network dimensions, namely *node*, *edge* (or *connection*) and *time*, can be manipulated to reduce clutter, thus leading to better layout readability and enhanced visual analysis.

For improving readability, some methods focus on obtaining a node ordering sequence that reduces the number of overlapping edges in the MSV layout by reducing their lengths (Figure 14(c)) (LINHARES et al., 2017b; ELZEN et al., 2013). Other studies consider different temporal resolutions in the network under analysis to reduce network data (Figure 14(e)). There are also methods that focus on removing less relevant edges, thus reducing the amount of visual information through edge sampling strategies (Figure 14(d)) (ZHAO et al., 2018). Despite these advances, improving the MSV readability still represents an open and challenging research problem (ZHAO et al., 2018).

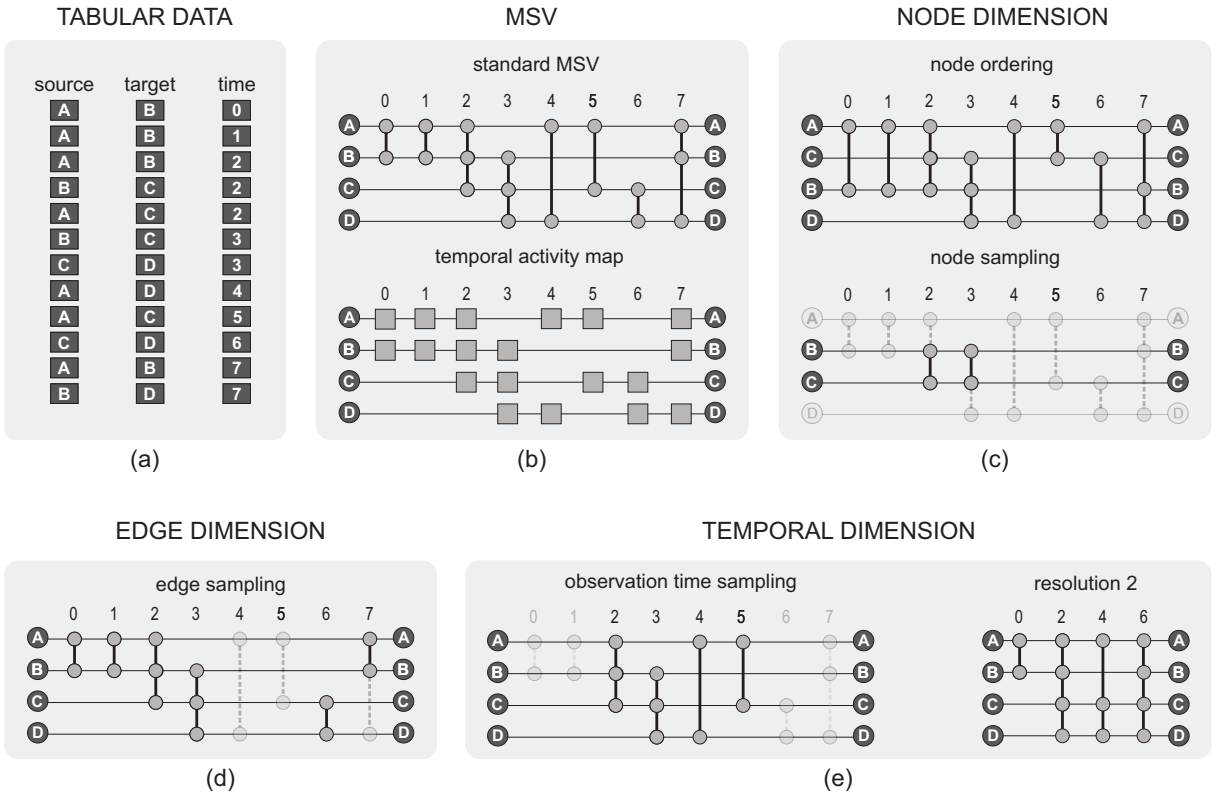

Figure 14 – MSV layout and possible types of manipulation. (a) Tabular (raw) data; (b) MSV layout: standard MSV, showing nodes and edges, and temporal activity map (TAM). In real-world networks that contain a large amount of data, MSV suffers from visual clutter. In this case, it is possible to improve the layout by manipulating the three network dimensions. (c) Node dimension manipulation – ordering and sampling; (d) Edge dimension manipulation – sampling; (e) Temporal dimension manipulation – observation time sampling and resolution change.

3.1.1 Node Ordering

Node ordering strategies are the best techniques to improve MSV (ZHAO et al., 2018). A naive node ordering strategy for MSV is the random placement of the nodes along the rows. Nodes can also be sorted (from top to bottom or vice versa) according to their order of appearance, i.e., according to the timings of first (or last) connection. This approach, called *Appearance* ordering, provides an easy way of identifying birth, death and lifetime of nodes (Figure 15(a)) (HOLME; LILJEROS, 2014). Moreover, it helps to identify if node and edge activities are spread all over the time period of the network or if they are concentrated in time (LINHARES et al., 2017b). Another ordering strategy, called *Lexicographic* ordering (ELZEN et al., 2013; LINHARES et al., 2017b), sorts nodes according to their labels or values, that may come from metadata such as classmates, age or same ward patients (Figure 15(b)). The main advantage of such strategy is to cluster nodes with similar features to facilitate cross-comparisons (LINHARES et al., 2019a).

These strategies explore activity patterns for ordering without taking into account visual clutter.

The are also more elaborate strategies, such as the *Optimized MSV* (ELZEN et al., 2013; ELZEN et al., 2014), a hierarchical strategy that combines the minimization of both edge block overlap and standard deviation of the edges sizes, reducing visual clutter and preventing unwanted visual attention. This approach, however, is not optimal for large networks, since it has visual scalability limitations due to the visual clutter generated when the number of nodes and/or edges is high (ELZEN et al., 2014). Another strategy is *Recurrent Neighbors (RN)* (LINHARES et al., 2017b), which spatially approximate nodes that are neighbors (adjacent nodes) on the layout (Figure 15(c)). In the first step, the node with the highest degree on the aggregated network (i.e., considering all timestamps at once) is positioned on the center of the layout. Next, the neighbors of this node with more connections to it are positioned around it on the layout. The ordering continues recursively by positioning closely the nodes that are more often connected together. RN presents the same limitations of Optimized MSV when applied to large networks.

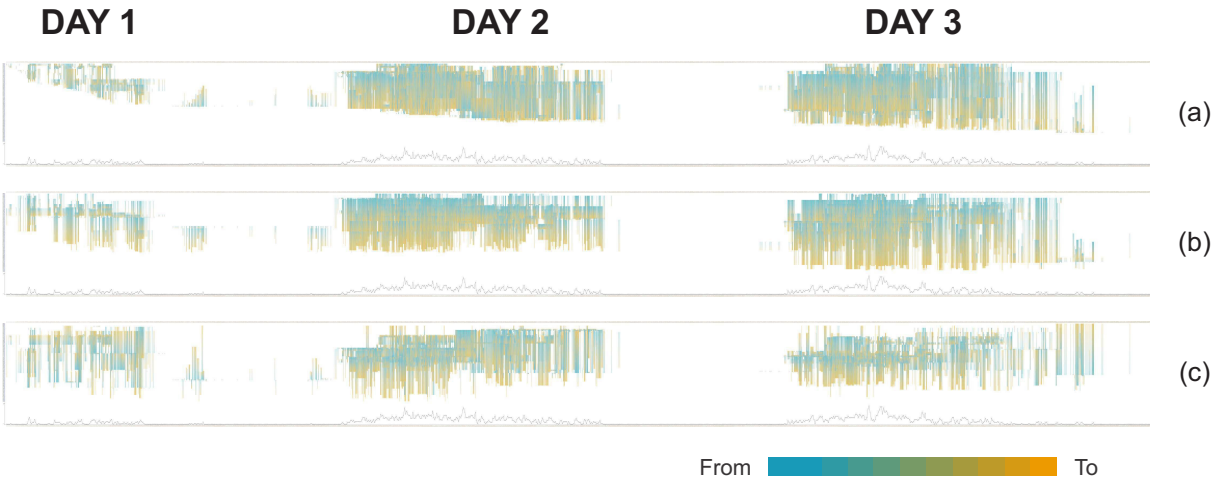

Figure 15 – MSV layouts generated by three node ordering methods for three days of the Hospital network (see Section 2.1.3). (a) *Appearance*; (b) *Lexicographic*; (c) *Recurrent Neighbors*. Adapted from (LINHARES et al., 2017a) ©2017 SBC.

3.1.2 Temporal Resolution

Besides node dimension, the temporal dimension can also be manipulated to improve layout readability (Figure 14(e)). One possibility is to choose an observation time of interest (e.g., only the first or the second day of the network), as adopted by Zhao et al. (2018) in a high-school network visual analysis. Another important manipulation in the temporal dimension is the change of the original network temporal resolution. In this case, edges from subsequent timestamps are grouped in a single timestamp (see Section 2.1.4).

For convenience, many studies change the original temporal resolution of the networks under analysis (LINHARES et al., 2017b; ZHAO et al., 2018; LINHARES et al., 2019a; ROCHA; MASUDA; HOLME, 2017). In (LINHARES et al., 2019a), for instance, the aforementioned high-school network was analyzed considering each time of the MSV layout as a three-minute interval whilst the original network temporal resolution is a 20-second interval per timestamp (MASTRANDREA; FOURNET; BARRAT, 2015). In the same way, the Enron network (see Section 2.1.3) was analyzed using a specific temporal resolution in (ZHAO et al., 2019) and a different one in (LINHARES et al., 2017b). All these studies, however, adopt a temporal resolution value that is chosen empirically, through initial exploratory analyses, or by a domain specialist that knows *a priori* which resolution scale is adequate for the analysis given the expected edge distribution. Regardless of the case, the adopted temporal resolution is a uniform, global and static value that may not faithfully represent the different levels of node activity, i.e., the number of edges and their distribution over time.

Wang et al. (2019) proposed a method that considers the level of node activity when changing the timestamp of the edges. Their strategy (hereafter named *Balanced Visual Complexity – BVC*) uses more timestamps to represent high-activity periods (with bursts of edges) and less timestamps otherwise. For this purpose, they adopt an approach similar to the histogram equalization, well-established in the discipline of digital image processing, to redistribute the edges in a way that creates an equal visual complexity, i.e., a balanced number of edges over time. As illustrated in Figure 16, patterns related to periods with burst of edges or inactivity (without edges) may be lost with BVC because of its equalization procedure. Moreover, BVC requires all edges in primary memory and so its application in streaming scenarios is infeasible without adaptation.

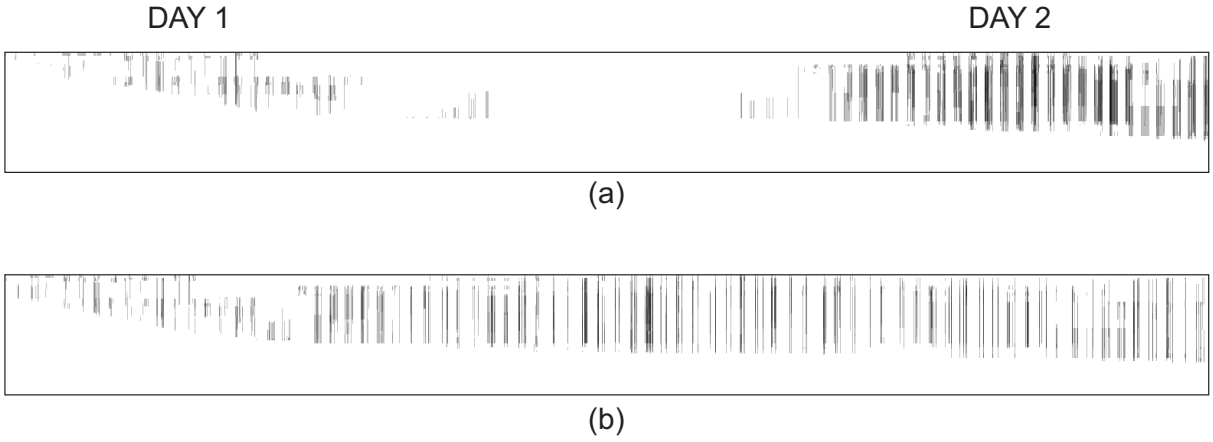

Figure 16 – *Balanced Visual Complexity (BVC)* applied to the Hospital network (see Section 2.1.3) using MSV. (a) MSV showing days 1 and 2 of the original network. (b) MSV showing days 1 and 2 after BVC. The identification of patterns related to periods with burst of edges or inactivity is impaired after BVC.

3.1.3 Edge Sampling

A naive strategy is the random sampling of edges over time. A more elaborated edge sampling method for improving MSV readability is the *Accept-Reject (AR)* sampling (ZHAO et al., 2018; ZHAO et al., 2019). It analyzes all possible node pair combinations in the temporal network to sample edges based on accept-reject random sampling, which accepts edges depending on a test involving the ratio of a target density distribution (that quantifies the edge distribution between a node pair) and a proposal distribution. AR is a non-deterministic method whose worst-case time complexity is $\mathcal{O}(m^2)$, where m is the number of edges. The characteristics of AR make its application in large networks or streaming scenarios infeasible.

An extension of the AR sampling was proposed by Zhao et al. (2018). The *Edge Overlapping Degree (EOD)* quantifies the overlap between neighboring edges and uses this value as an edge-level indicator of visual clutter. Probability density functions based on kernel density estimation are then used to perform the sampling, which also takes into account the length of the edges as longer edges tend to cause more visual clutter than shorter ones. EOD has the same worst-case time complexity of AR and presents the same limitations. Furthermore, EOD cannot be applied in layouts other than MSV without adaptation. According to the EOD creators, no previous studies developed edge sampling strategies for improving MSV readability (ZHAO et al., 2018).

3.2 Manipulating Dimensions in Streaming Networks

In this section, we present existing methods that manipulate streaming network dimensions. In temporal network manipulation, subject of the previous section, all edges and nodes are available. In streaming networks, on the other hand, the distribution of incoming data is non-stationary. This characteristic increases complexity and makes streaming network manipulation even more challenging.

3.2.1 Node Positioning

The size of streaming networks, in terms of number of nodes, edges, and timestamps, is potentially unbounded. Since the entire network does not fit in primary memory and, consequently, on the screen, the visualization requires animation or the exhibition of selected timestamps as small multiples (CRNOVRSANIN; CHU; MA, 2015), both typically using node-link diagrams. For temporal networks, in which all nodes and edges are available, one can optimize the layout for animation, finding the node positioning[1] that produces the layout with less visual clutter (CRNOVRSANIN; CHU; MA, 2015). However, in

[1] For node-link diagrams, we adopt the term *positioning* instead of *ordering* because there is no explicit sorting of nodes as occurs in the MSV layout.

streaming scenarios, the distribution of future data is non-stationary. This characteristic increases streaming network visualization complexity.

In (FRISHMAN; TAL, 2008), the authors proposed a force-directed algorithm that incrementally defines node positioning while taking into account the user's mental map and the global network structure (Figure 17). The algorithm controls node displacement over time by allowing recently updated nodes (nodes whose positions were changed) and their neighbors (adjacent nodes) to move. Each node receives a weight, called pinning weight, that determines its range of movement. To evaluate a layout quality, the authors introduced the notion of potential energy. Lower energy leads to shorter edges, which leads to less overlaps.

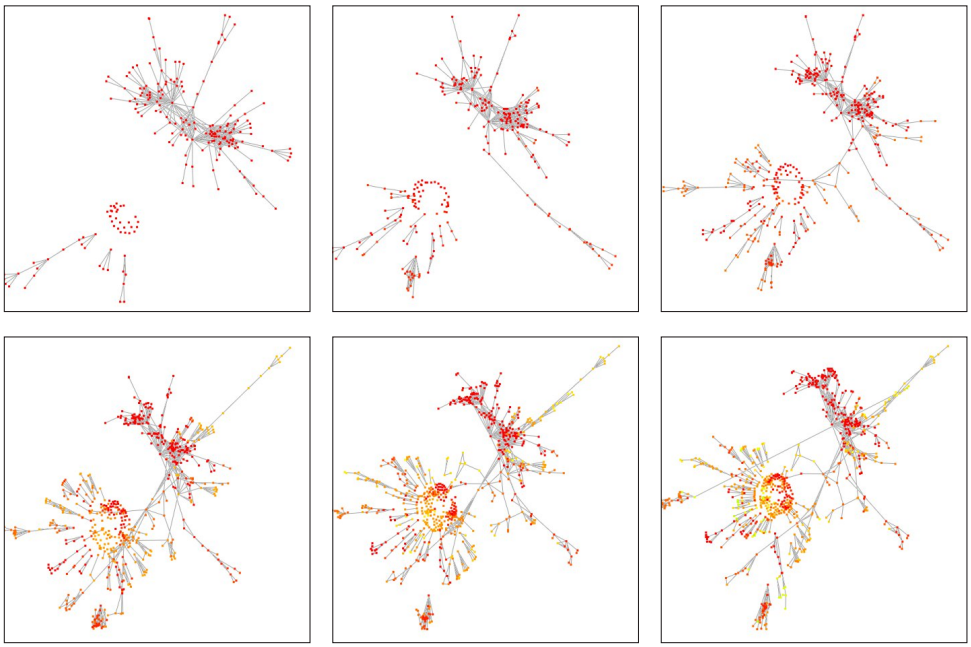

Figure 17 – Evolution of a social network with node positioning defined by (FRISHMAN; TAL, 2008). The network evolves from left to right and from top to bottom. Nodes are colored by age using a color scale composed of red (old nodes), yellow, and green (recent nodes). Adapted from (FRISHMAN; TAL, 2008) ©2008 IEEE.

In (GOROCHOWSKI; BERNARDO; GRIERSON, 2012), the authors developed a streaming node positioning method that employs the concept of node aging. The age of a node is calculated based on when the node appeared and in the amount of local movements that occurred in its neighborhood. The idea is that younger nodes have a large range of movement while older ones reduce their movements, thus maintaining unchanging regions with fixed positions on space. In streaming scenarios, animated age-directed layouts reduce unnecessary node movement (GOROCHOWSKI; BERNARDO; GRIERSON, 2012).

Another streaming node positioning method is presented in (CRNOVRSANIN; CHU; MA, 2015). The authors proposed a streaming method that defines node positioning

based on nodes' attractive and repulsive forces. The basic idea is to reduce long edges by approximating the nodes until they reach a low state energy. By reducing edge lengths and edge crossings, the layout readability is improved. Figure 18 presents the evolution of the Facebook network (VISWANATH et al., 2009), from time 610 to time 680, according to this node positioning method.

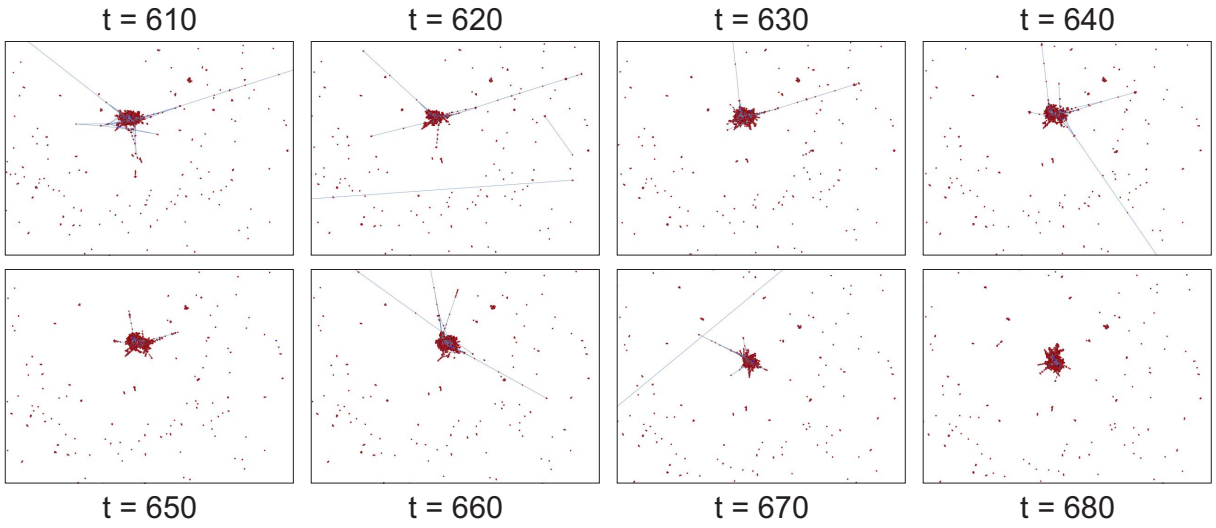

Figure 18 – Evolution of the Facebook network (VISWANATH et al., 2009), from time 610 to time 680, with node positioning defined by (CRNOVRSANIN; CHU; MA, 2015).

Besides the node positioning quality, some other factors influence layout readability. The major issue is the choice of the animation transition speed (REY; DIEHL, 2010; BECK et al., 2017). Nevertheless, strategies for reducing visual clutter in node-link diagrams are generally hopeless even if combined with animated graphs because it is difficult to maintain the mental map representation of the layout for networks that change in time (LINHARES et al., 2019a; ARCHAMBAULT; PURCHASE, 2016).

3.2.2 Temporal Resolution

As discussed in Section 3.1.2, the temporal resolution adopted in temporal network analyses is a global and static value that may not correspond to the different levels of node activity over time. In temporal networks, however, all nodes and edges are available, so different resolution values may be tested until an adequate one is found.

In streaming scenarios, although one can adopt a global and static temporal resolution as well, the choice of its value is even more difficult due to the non-stationary distribution of future edges. Exploratory analysis may not be possible because usually there are no *a priori* data to support the decision. Since the edge distribution can change, considering an initial set of nodes and edges of the stream to support the choice may be inefficient as

well. This is often ignored when it is assumed a uniform edge distribution, as if the edges came in consecutive timestamps (CRNOVRSANIN; CHU; MA, 2015; AHMED et al., 2017). The only method that we are aware of that considers local levels of node activity over time is BVC (see Section 3.1.2), but it is not suitable for streaming networks. To the best of our knowledge, no one has proposed streaming methods that change the temporal resolution taking into account varying levels of node activity over time.

3.2.3 Edge Sampling

The naive random edge sampling over time, suitable for temporal networks, is also suitable for streaming networks. Despite this naive strategy, one of the first attempts to sample streaming data in a single pass (linear computational time complexity) was the *Random Reservoir Sampling* (VITTER, 1985). It works by defining a reservoir of size k and adding the first k stream elements in that reservoir. Then, each new element (the i^{th} element, $i > k$) has a uniform probability (k/i) of being accepted and replace a randomly selected reservoir element (GAMA, 2010). Reservoir sampling, when applied to streaming networks, leads to a uniform edge sampling (JHA; SESHADHRI; PINAR, 2015). The same idea was used to create *Partially Induced Edge Sampling (PIES)* (AHMED; NEVILLE; KOMPELLA, 2013), a method that combines sampling with partial network induction in a single pass execution. Initially, the first m streaming edges incident to n nodes are inserted in the edge reservoir and these n nodes are inserted in the node reservoir. Then, for each incoming edge, the method (probabilistically) includes its incident nodes in the reservoir by replacing randomly selected reservoir nodes. Along with the node reservoir update, PIES also adds every edge whose incident nodes already belong to the reservoir – a step responsible for the partial induction. PIES is more suitable for sparse and less clustered streaming networks (AHMED; NEVILLE; KOMPELLA, 2013). Besides, PIES was not tested in visualization tasks.

A sampling method that discards less connected nodes of a given streaming network was proposed in (SARMENTO; CORDEIRO; GAMA, 2015b). The method adopts Landmark Windows (see Section 2.3) and uses *Space-Saving* (METWALLY; AGRAWAL; ABBADI, 2005), an incremental method that maintains a list of the top-k most frequent items in a stream (GAMA, 2010). As the network evolves, nodes that enter in the top-k list are added to the sample along with their edges. In the same way, nodes that exit the list are removed along with their edges. This method was applied to streaming network visualization using node-link diagrams (SARMENTO; CORDEIRO; GAMA, 2015a). The authors also used the top-k nodes and their first and second neighborhood-order connections to perform community detection and showed that the network communities from the sampled network match those from the original network (SARMENTO; CORDEIRO; GAMA, 2015b).

Sikdar et al. (2018) developed *Community Preserving sampling Algorithm for Stream-*

ing graphs (ComPAS), a streaming sampling algorithm that also preserves the community structure of the original network in the sample. The method identifies nodes with high degree and high clustering coefficient and determines in which community they should be inserted through an incremental modularity computation. ComPAS has (almost) linear time complexity (SIKDAR et al., 2018) and was not tested in visualization tasks as well.

Different objectives guide the development of sampling methods for streaming networks. Some strategies are focused in triangles estimation – used for clustering coefficient computation (ETEMADI; LU, 2019) –, outlier detection (AGGARWAL; ZHAO; YU, 2011), or feature preservation (SIKDAR et al., 2018). Regardless of the motivation, network sampling remains an open research field (SARMENTO; CORDEIRO; GAMA, 2015b).

3.3 DyNetVis

Several computational tools have been proposed to visualize complex networks, including some libraries for R and Python (programming languages), Gephi (BASTIAN; HEYMANN; JACOMY, 2009), and Cytoscape (SHANNON et al., 2003). Some of them, such as the aforementioned Gephi and the Cytoscape plugin DyNetViewer (LI et al., 2017), also present functionalities for visualizing networks with temporal information through animated – or small multiples over – node-link diagrams (for details about these visualizations, please refer to Section 2.2.1). There are also softwares focused on computations involving large networks (e.g., GraphChi (KYROLA; BLELLOCH; GUESTRIN, 2012) and Pregel (MALEWICZ et al., 2010)), but none of them considers network visualization.

DyNetVis[2] (LINHARES et al., 2017b) is a freely available software for visualizing temporal networks that implements all three layouts mentioned and used in this thesis: node-link diagram, MSV, and TAM. It also provides all node ordering methods presented in Section 3.1.1 (except *Optimized MSV*) and allows changes in the network temporal resolution.

Several interactive tools are provided by DyNetVis. Examples include changing the color of nodes (MSV and TAM) and edges (only MSV) to represent structural properties (e.g., node degree), metadata (e.g., age or gender), and others (see Figures 19(b-c)). Users can also analyze particular regions of interest by selecting all edges that connect specific nodes (Figure 19(d)), nodes and edges from specific timestamps (Figure 19(e)), or arbitrary nodes and edges (Figure 19(f)). Other interactive tools, such as zoom and pan, are also available. Not least, DyNetVis also provides *coordination* between layouts. With this feature, nodes and edges selected in a layout are automatically selected in the others (Figure 20).

[2] <www.dynetvis.com>

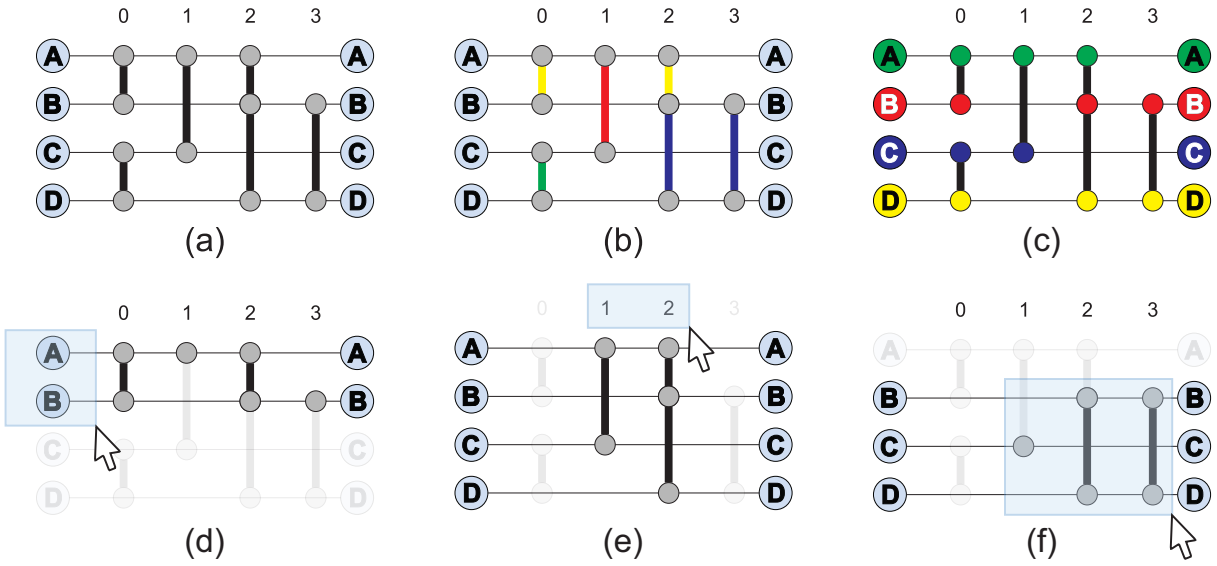

Figure 19 – Some of the interactive tools provided by DyNetVis for MSV. (a) Original layout. (b) Using color to represent an edge attribute. (c) Using color to represent a node attribute. (d) Selection of all edges involving specific nodes. (e) Selection of all edges and nodes occurred in specific timestamps. (f) Selection of specific nodes and edges. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

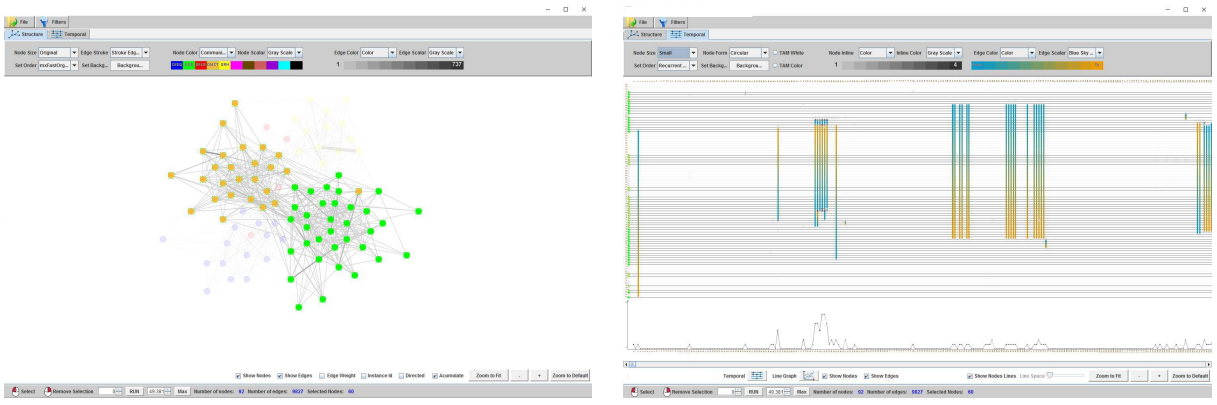

Figure 20 – DyNetVis. (a) Node-link diagram and its specific interface options. (b) MSV and its specific interface options. Through coordination between the layouts, nodes and edges selected in one of them are automatically selected in the other. Adapted from (LINHARES et al., 2017a) ©2017 SBC.

DyNetVis was developed by our research group and is currently open-source. Some of the methods that we propose in this thesis are available in the most recent version of the software; the others will be included in new versions. All visual analyses in the case studies presented in the next chapters were performed using DyNetVis.

3.4 Final Considerations

This chapter presented how existing studies manipulate *node*, *time*, and *edge* dimensions for temporal and streaming networks (see Table 1). MSV better preserves the user's mental map of the network evolution in comparison to the node-link diagram, even when using animation (LINHARES et al., 2019a). In this sense, we presented studies that use MSV whenever possible. Moreover, node ordering strategies are the best techniques to improve MSV (ZHAO et al., 2018), so the studies related to the node dimension that we presented for MSV focus on node ordering strategies instead of node sampling. This chapter also introduced DyNetVis, an interactive system for visualizing temporal networks that is being extended to incorporate our proposals.

Table 1 – Methods discussed in this chapter.

Dimension	Method	Temporal Networks	Streaming Networks
Node	Random	✓	✓
	Appearance	✓	
	Lexicographic	✓	
	Linhares et al. (2017b)	✓	
	Elzen et al. (2014)	✓	
	Frishman e Tal (2008)		✓
	Gorochowski, Bernardo e Grierson (2012)		✓
	Crnovrsanin, Chu e Ma (2015)		✓
Time	Static Resolution	✓	✓
	Wang et al. (2019)	✓	
Edge	Random	✓	✓
	Zhao et al. (2019)	✓	
	Zhao et al. (2018)	✓	
	Ahmed, Neville e Kompella (2013)		✓
	Sarmento, Cordeiro e Gama (2015b)		✓
	Sikdar et al. (2018)		✓

In this thesis, we propose methods for streaming networks that manipulate the network dimensions to reduce visual clutter and thus enhance the visual analysis. We evaluated our methods mainly using the MSV layout, but the majority of our proposals could be used in animated node-link diagrams or other layouts as well. In the next chapter, we introduce our adaptive temporal resolution method and show that changing the temporal resolution according to local levels of node activity enhances network visualization.

Chapter 4

Temporal Dimension

In this chapter, we propose a method that automatically adapts the temporal resolution scale of a network, according to the different levels of node activity over time. It runs in a streaming-fashion and gives more importance to recent edges. Our resolution approach allows the identification of patterns, based on local levels of activity over time, that would be lost or difficult to find with a global and static resolution scale.

Researchers from Network and Visualization communities may sometimes adopt different terminologies. Visualization experts may associate "changing the temporal resolution" with "timeslicing", and this can be performed by considering local levels of node activity over time (nonuniform timeslicing) or not (uniform timeslicing). In the same way, streaming methods are referred as online methods. The terminology used in this thesis was chosen in an attempt to provide a broad understanding among readers from different communities.

4.1 Adaptive Temporal Resolution

Each timestamp in a temporal network comprehends a time interval defined by the network temporal resolution (see Section 2.1.2). The idea behind our adaptive temporal resolution method is that same duration intervals that have different levels of node activity must be represented by different numbers of timestamps. Higher levels of node activity leads to fewer timestamps. In this way, the amount of information shown in this portion of the layout is reduced, favoring the identification of patterns.

Our technique considers the level of node activity, i.e., the number of edges and their distribution, on a fixed size sliding window of timestamps (window of size w_{size}) to decide the temporal resolution scale to be used in the next window. Inside a window, the edge distribution is considered by reducing the importance of older edges using the forgetting mechanism *fading sum* (GAMA, 2010; GAMA; SEBASTIÃO; RODRIGUES, 2013).

Initially, we adopt the original resolution scale in the first window (cold start). From there, the resolution value for each subsequent window is calculated according to Equa-

tion 2:

$$\sigma_n = \lfloor \delta.\sigma_c + (1 - \delta).f_s(w_{size}) \rfloor \tag{2}$$

where σ_n is the resolution value for the next window, δ ($0 \leq \delta \leq 1$) is a constant that determines the importance of the current resolution value (σ_c) in the computation of the new resolution, and $f_s(w_{size})$ is the fading sum of all edges in the current window, which is calculated according to Equation 3:

$$f_s(i) = \frac{x_i}{t_{wc}} + \alpha.f_s(i-1) \tag{3}$$

where x_i is the number of edges in position i of the window, t_{wc} is the number of timestamps considered by the window that presents at least one edge, $f_s(1) = \frac{x_1}{t_{wc}}$, and α ($0 \ll \alpha \leq 1$) is the fading factor. If $\sigma_n = 0$, then σ_n is set as the average value of all past resolutions, so large inactivity periods (i.e. without edges) may be represented by a resolution that is different from the original.

With the new resolution scale computed, it is possible to change the timestamp attribute of the incoming edges. Inspired by Equation 1 (Section 2.1.4), we define the new timestamp t_{new} of a connection e as:

$$t_{new}(e) = \left\lfloor \frac{t_{orig}(e) - t_{ini}}{\sigma_n} \right\rfloor + t_{ref} \tag{4}$$

where $t_{orig}(e)$ is the timestamp of e in the original resolution, t_{ini} is the first timestamp considered by the current sliding window, σ_n is the new resolution value (Equation 2), and t_{ref} is the timestamp that acts as a reference in order to apply the resolution scale in inactive timestamps. The value of t_{ref} is computed when dealing with the first edge of a new resolution and is defined according to Equation 5:

$$t_{ref} = \left\lfloor \frac{t_{ini} - t_{orig}(e')}{\sigma_p} \right\rfloor + t_{new}(e') \tag{5}$$

where $t_{orig}(e')$ is the original timestamp of the last edge from the previous window (edge e'), σ_p is the previous resolution (applied on e'), and $t_{new}(e')$ is the timestamp of e' in σ_p.

Figure 21 shows an example of how the timestamp of an edge is changed according to Equation 4. In this figure, $t_{orig}(e') = t_{new}(e')$ because, up to this point, the original resolution (value 1) was maintained in the network. In $t = 100$, the new resolution value was computed (value 2) and so it was necessary to change the original timestamp of e. Each timestamp in resolution 2 is twice the time interval represented by a timestamp from resolution 1, thus $t_{orig}(e) = 130$ and $t_{new}(e) = 115$. Assuming an edge x with $t_{orig}(x) = 131$, then $t_{new}(x)$ would be equal to 115 as well, and so on. As stated, Equation 4 takes into account inactivity periods, respecting their occurrence in the converted timestamps.

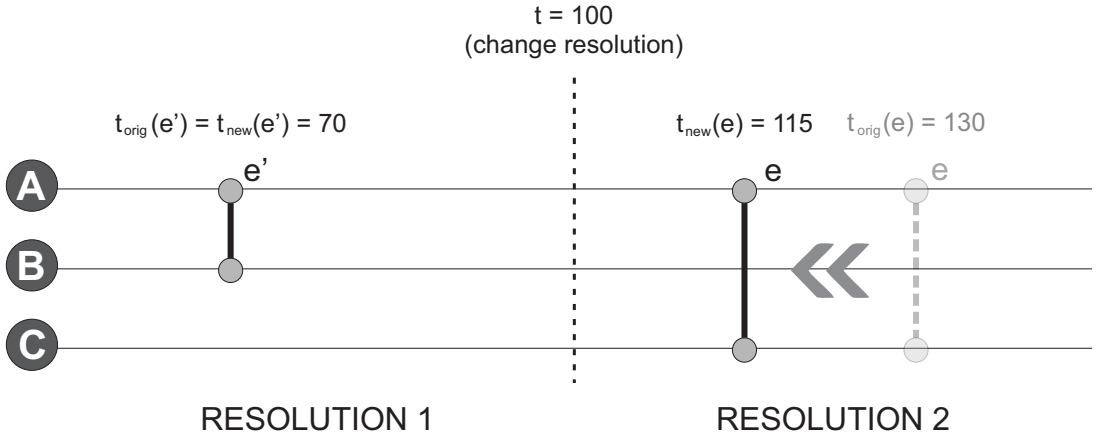

Figure 21 – Example of an edge timestamp change due to the new resolution value. In resolution 2, the timestamp of e is changed from 130 to 115.

4.2 Case Studies

In this section, we present a visual analysis of two real-world temporal networks in order to compare our adaptive resolution against the original scale. For this purpose, we consider these networks as streaming networks, using them to simulate continuous arrival of edges. We consider resolution 1 (Res. 1) as the original resolution of the network and $\delta = 0.2$ (empirically determined) as the importance of the current resolution in the computation of the new one (see Equation 2). All experiments were performed using DyNetVis, which implements the adaptive temporal resolution method and all layouts and features presented in this section. For details about DyNetVis, please refer to Section 3.3.

4.2.1 Primary School

This analysis considers the *Primary School* network, described in Section 2.1.3. Figure 22 presents the TAM layout for four classes and all teachers of the Primary School network in resolution 1 (original). The nodes are grouped according to the classes and grades. The layout is horizontally large (due to the number of timestamps), which impairs the identification of global patterns and requires more screen space and scrolling, which impairs the user's perception of temporal changes during the network evolution (mental map preservation (PURCHASE; SAMRA, 2008)). Moreover, the layout is dense (a lot of edges over time) and only a few patterns are easily identified, as, for example, the absence of classes 4A and 4B students near the end of the second day. The network does not register contacts during sports activities (STEHLÉ et al., 2011), so it is possible to assume that these classes were involved in such activities or dismissed. Another possibility is that the students were taking exams or other activities without interacting with each other.

To improve pattern identification, we applied our adaptive resolution in the Primary School network. Different values for Fading Factor (FF) and window size (w_{size}) were

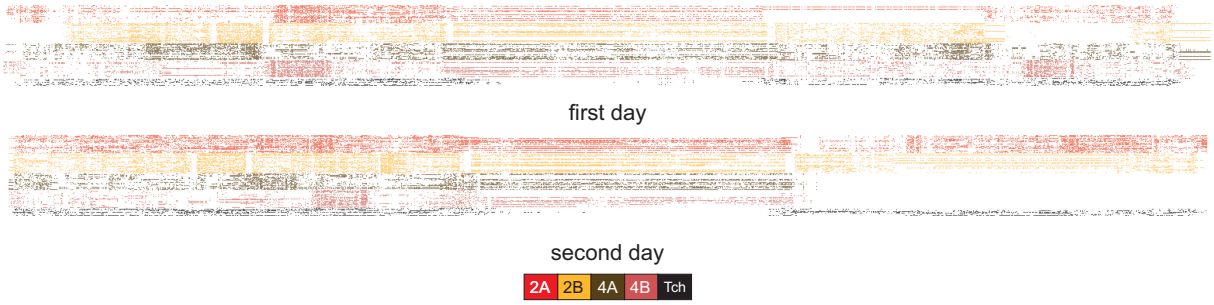

Figure 22 – TAM layout showing four classes and all teachers of the Primary School network using resolution 1 (original). The interval between both days (from 5.21pm to 8.29am) does not present any connection and was omitted due to its size in the layout. The nodes are grouped according to the classes and grades. The "Tch" profile refers to the teachers of the school. The layout is dense and has few visible patterns, as, for example, the absence of classes 4A and 4B near the end of the second day.

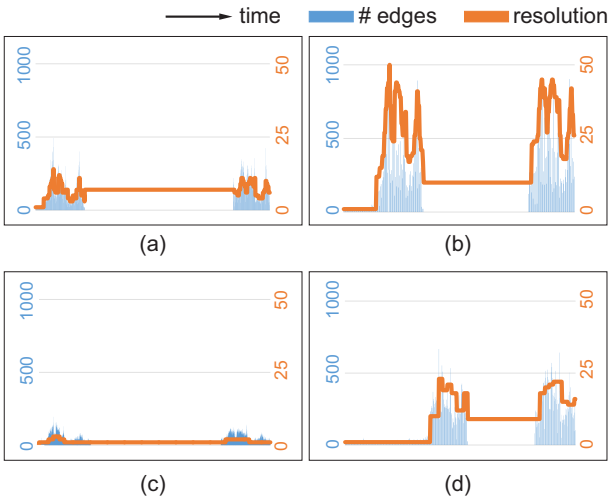

Figure 23 – Adaptive resolution and its relation with the edge distribution for the Primary School network. (a) $w_{size} = 50$ and $FF = 0.9$ (1,443 timestamps). (b) $w_{size} = 50$ and $FF = 0.99$ (353 timestamps). (c) $w_{size} = 200$ and $FF = 0.9$ (4,880 timestamps). (d) $w_{size} = 200$ and $FF = 0.99$ (541 timestamps). The choice of the Fading Factor (FF) and the window size (w_{size}) affects the resolution scale and, consequently, the layout and visible patterns.

evaluated and their impact in the resolution change is shown in Figure 23. In the figure, the whole network is considered (activity in day 1, interval, activity in day 2). By comparing the plots in which the FF value is the same ($FF = 0.9$ in (a,c) and $FF = 0.99$ in (b,d)), it is possible to see that a large window makes that the perception of changes in the level of node activity be late, delaying the resolution change. As a consequence, patterns related to these changes may be lost or identified only many timestamps later. This is especially relevant in streaming network analysis, in which the past data may have already been discarded. By comparing the plots in which the w_{size} is the same ($w_{size} = 50$

in (a,b) and $w_{size} = 200$ in (c,d)), one can notice higher resolution values when adopting higher FF. This is expected since high FF values increases the importance of old edges. Finally, the plots show the resolution adopted in the interval between both days of the network, in which there is no edge. The value is computed based on the average value of the past resolutions. The choice of this value instead of the original resolution is due to the space of the layout required to represent such interval, which would be many times greater in the original resolution.

Figure 24 shows TAM layouts for different temporal resolutions considering the same four classes and teachers from Figure 22. Figure 24(a-e) shows TAM layouts for the static resolutions 10, 25, 39, 100, and 200, respectively. Figure 24(f) shows the TAM layout generated by our adaptive resolution method ($w_{size} = 100$ and $FF = 0.99$, chosen empirically). Resolutions 10, 25, and 39 (Figure 24(a-c)) are shown because they represent the lower, the average and the higher resolution values adopted by our method for this network. Resolutions 100 and 200 (Figure 24(d-e)) are arbitrary values. As expected, higher resolution values generate denser and (horizontally) smaller layouts, which impairs the visual analysis and the identification of patterns. On the other hand, our adaptive resolution method founds resolution values that represent appropriate levels of visual density.

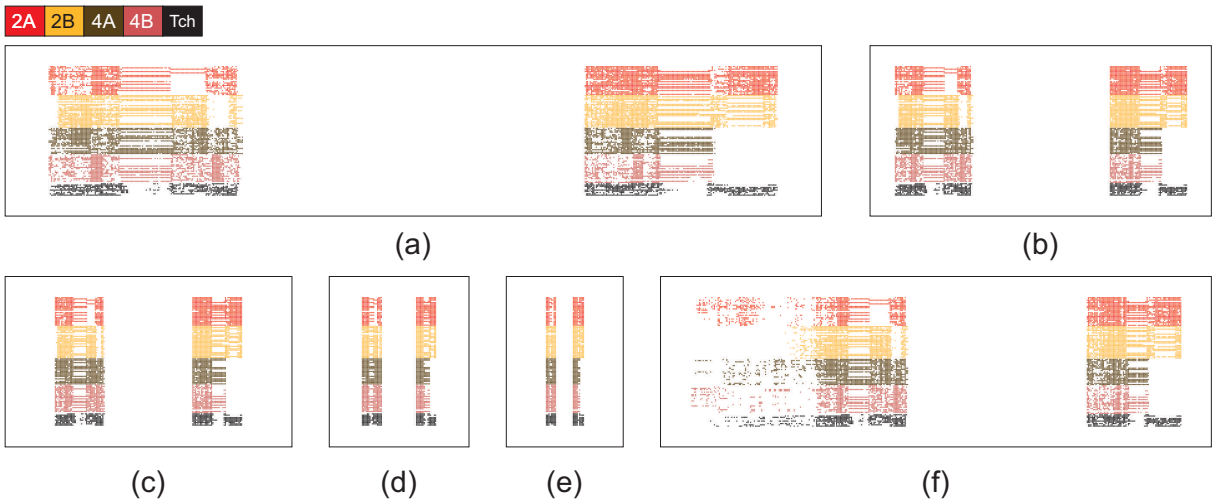

Figure 24 – TAM layouts showing four classes and all teachers of the Primary School network for different temporal resolutions. (a) Res. 10. (b) Res. 25. (c) Res. 39. (d) Res. 100. (e) Res. 200. (f) Adaptive resolution adopting $w_{size} = 100$ and $FF = 0.99$.

The choice of the temporal resolution highly affects pattern identification. Figure 25 presents visual analyses over the TAM layouts generated by the adaptive resolution (adopting $w_{size} = 100$ and $FF = 0.99$, Figure 25(a)) and by static resolutions 25 and 200 (Figure 25(b,d), respectively). These are the same layouts from Figure 24(f,b,e). The layout generated by BVC is also considered (Figure 25(c)). In the best-case scenario, at

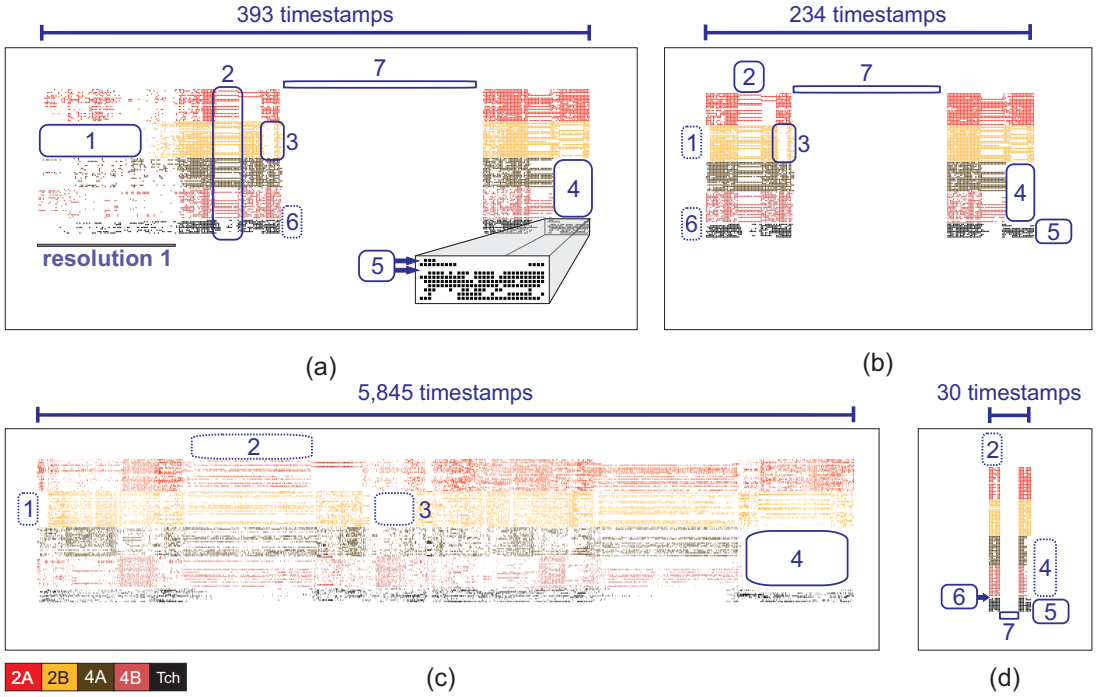

Figure 25 – TAM layouts generated by different temporal resolutions and their visible patterns for the Primary School network. (a) Adaptive resolution adopting $w_{size} = 100$ and $FF = 0.99$. (b) Res. 25. (c) BVC. (d) Res. 200. A maximum of seven patterns can be identified: (1) class 2B students joined the network after the others; (2) lunch break; (3) no interaction involving class 2B students near the end of the 1^{st} day; (4) absence of classes 4A and 4B near the end of the 2^{nd} day; (5) two teachers left the network after lunch in the 2^{nd} day; (6) some students did not join the network in the 1^{st} day; and (7) inactivity period. Continuous rectangles represent patterns considered as easy to identify. Dotted rectangles represent patterns with difficult perception.

least seven patterns can be identified: (1) all students from class 2B joined the network after the other classes and the group of teachers; (2) lunch break – several students go home for lunch, which reduces the number of interactions in such interval (STEHLÉ et al., 2011); (3) there is no interaction involving class 2B students in a time interval near the end of the first day – probably due to sports activities (STEHLÉ et al., 2011); (4) absence of classes 4A and 4B students near the end of the second day; (5) two teachers left the network after lunch in the second day – probably the teachers from classes 4A and 4B; (6) there are students that did not join the network in the first day; and (7) inactivity period due to the absence of classes (from 5.21pm to 8.29am).

Our adaptive method allows the identification of all seven patterns (Figure 25(a)), being five of them easily identified (1-5, 7). Pattern 6 is harder to identify because of the method's cold start (adoption of resolution 1 at the beginning of the layout), which pollutes the layout and impairs the perception of this pattern. Although this original resolution serves only as a start point, considering it inside the adaptive resolution layout facilitated the perception of pattern 1. This pattern can also be noticed when adopting

only resolution 1 in the analysis (see Figure 22), but not as fast as with our proposal. Patterns 2, 5, and 6, on the other hand, cannot be identified with resolution 1 (Figure 22). By adopting the static resolution 25 (Figure 25(b)), all seven patterns can be identified as well, five of them being considered as easy to found (2-5, 7) and two of them being a little harder (1,6). Although this layout allows the identification of all patterns, recall that this resolution is the average value considered by our adaptive method, which supports our method's quality. When considering BVC in the network analysis, one may see the edge redistribution caused by BVC's histogram equalization. As a consequence, pattern 7 is lost. Due to the number of timestamps, patterns 5-6 are also lost and patterns 1-3 are difficult to perceive. Only pattern 4 is considered easy to found. Such pattern, however, is also easy to identify with our method and with static resolutions 1 and 25. By using resolution 200 (Figure 25(c)), patterns 1 and 3 are lost and only patterns 5 and 6 are easily identified. Note that pattern 6 is more easily perceived in this layout, so higher resolution values may be useful in specific scenarios as well.

In summary, our method reduces the amount of visual information to an appropriate level that optimizes the identification of global patterns that are lost or difficult to perceive with other strategies, including BVC. Our layout for this network spent 393 timestamps (against 5,845 from BVC) while preserving all seven analyzed patterns. Less timestamps leads to less screen space and decreases the need of (horizontal) scrolling, which tends to facilitate the perception of temporal changes in the network (better mental map preservation). Considering the static resolutions, one should test different resolution scales until the better one is found. This approach, however, is only possible when dealing with (non-streaming) temporal networks. Our method not only provides adequate resolution scales, but is suitable for streaming scenarios in which edges are continuously arriving in non-stationary distribution.

Figure 26 shows the spread of edges over time according to different resolutions: the original resolution, BVC, our method ($w_{size} = 100$ and $FF = 0.99$), and Res. 25. The absence of edges in the middle of plots (a,c,d) corresponds to the inactivity period between both days of the network. While BVC (Figure 26(b)) changes the edge distribution because of its histogram equalization – which impacts pattern identification –, our method (Figure 26(c)) provides a distribution similar to those from static approaches (Figure 26(a,d)). Since our method adopts the original resolution in the first window (cold start), one may see a "shift" in the time dimension at the plot (Figure 26(c)).

Figure 27 shows the empirical cumulative distribution function (ECDF) considering the edges from our method's layout ($w_{size} = 100$ and $FF = 0.99$, Figure 27(a)) and from resolution 1's layout (Figure 27(b)). Our method produces less timestamps without edges when compared with the original resolution (36.6% vs 47% – blue dotted lines), which is justified by the resolution scale used in inactivity periods, that is different from the original one. Furthermore, 25.4% of the time contains very few edges in our layout

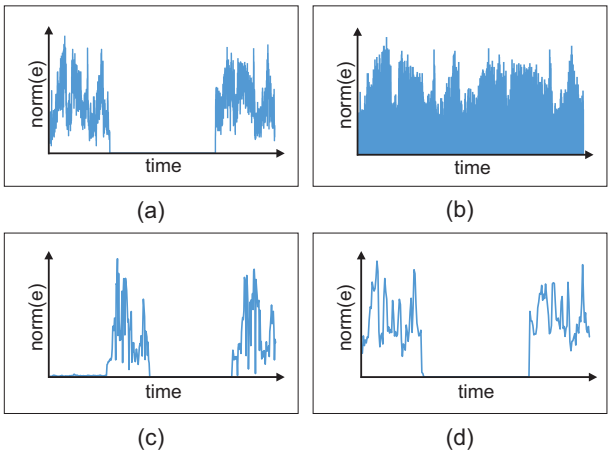

Figure 26 – Spread of edges according to different resolution scales for the Primary School network. (a) Res. 1. (b) BVC. (c) Our method ($w_{size} = 100$ and $FF = 0.99$). (d) Res. 25. "norm(e)" refers to the normalization of the number of edges to values between 0 and 1.

(cold start window). By observing the third quartile (red dotted lines), after 75% of the time our layout contains a maximum of 213 edges per timestamp (26% of the maximum number of edges per timestamp), while in resolution 1 the number of edges per timestamp is almost 43% of the maximum number of edges per timestamp (40 out of 94 edges).

The visual analysis can be performed from a different perspective by showing only edges, as illustrated in Figure 28, that shows the interactions involving classes 2A, 2B, and 4A over a MSV layout generated with the adaptive resolution method ($w_{size} = 100$ and $FF = 0.99$). This layout reaffirms: (i) students from class 2B joined the network after the others; (ii) students from class 4A left the network earlier than the others in the second day; (iii) the absence of the majority of 2A students, as well as 2B students, during a period after lunch in the first day. Besides, this layout reveals new patterns, such as the perception that the only two students from class 2A that stayed in the network during the time interval after lunch in the first day connected to one another. Moreover, the layout shows that students from one class have few interactions with students from other classes, with the majority of these interactions occurring during lunch. Not least, students from the 2^{nd} grade interact more between themselves than with class 4A. This behavior is also observed in the rest of the network (a lot of connections among students of the same grade and few connections involving different grades). These situations are expected in the network (STEHLÉ et al., 2011) and easily perceived in this layout.

4.2.2 Enron

This analysis considers the *Enron* network, described in Section 2.1.3. Enron was analyzed by several studies in literature, from non-streaming visual analysis using MSV (e.g., (LINHARES et al., 2017b; ELZEN et al., 2013; ZHAO et al., 2018)) to streaming-

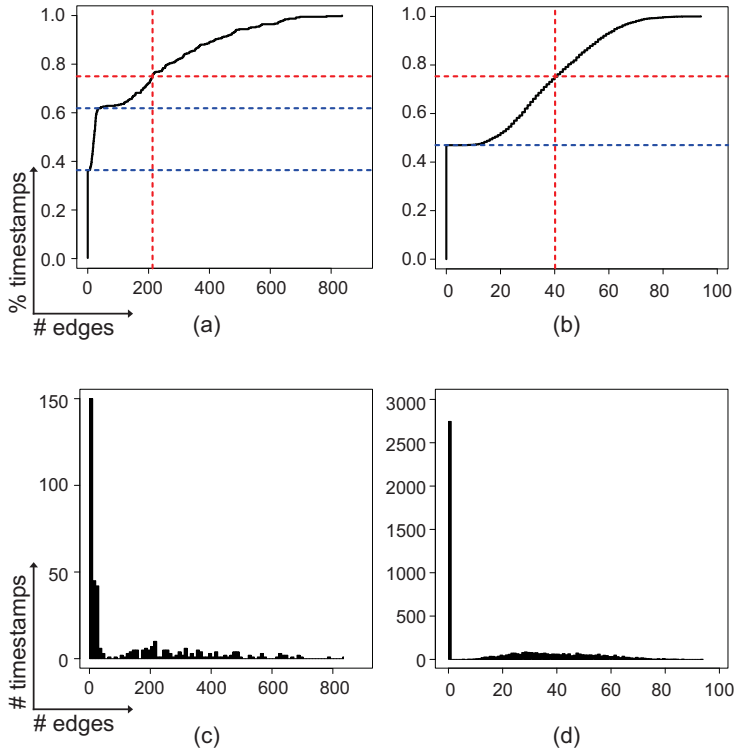

Figure 27 – Empirical cumulative distribution function (ECDF) and edge distribution (ED) considering the edge from the Primary School network. (a) ECDF our method ($w_{size} = 100$ and $FF = 0.99$). (b) ECDF Res. 1. (c) ED our method (393 timestamps, $w_{size} = 100$ and $FF = 0.99$). (d) ED Res. 1 (5,846 timestamps).

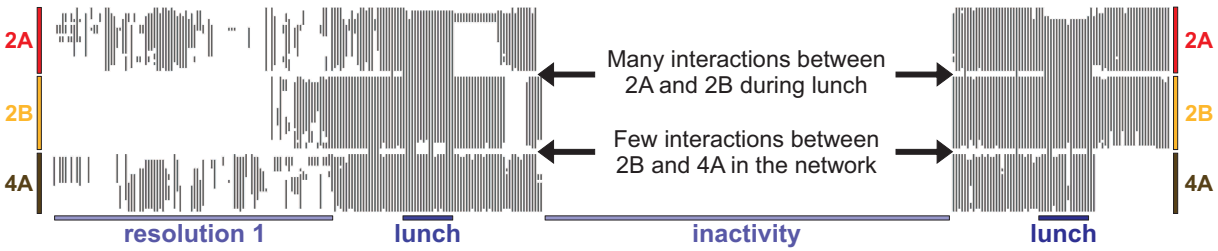

Figure 28 – MSV layout with adaptive resolution ($w_{size} = 100$ and $FF = 0.99$) showing connections between classes 2A, 2B and 4A. This layout reaffirms some of the patterns previously described and allows the identification of new ones.

fashion mining tasks (e.g., (SUN et al., 2007)). Unlike the Primary School network, whose level of node activity varies a lot in each day and which contains a large time interval without any edges (the period between the two days), the Enron network presents a growing level of node activity over time. We applied our adaptive method in the network to analyze the evolution of the resolution under this circumstance.

Figure 29 presents the adaptive resolution behavior under different values of w_{size} and FF for the Enron network. Comparing vertically the plots (a,c), it is possible to see the impact of the fading factor in the resolution computation. As can be seen, the high level

of activity near the end of the network is reflected in the resolution for the two FF values tested. Comparing the plots (b,c,d), one can see how frequent the resolution change occurs according to the window size. As discussed, large windows make the resolution change less frequent and, as a consequence, each resolution value may not faithfully represent the different levels of node activity. One can see such situation occurring in the Enron network by analyzing the resolution evolution under $w_{size} = 200$ and $FF = 0.99$ (Figure 29(d)): at the end of the network, the number of edges decreases abruptly, but the resolution remains high.

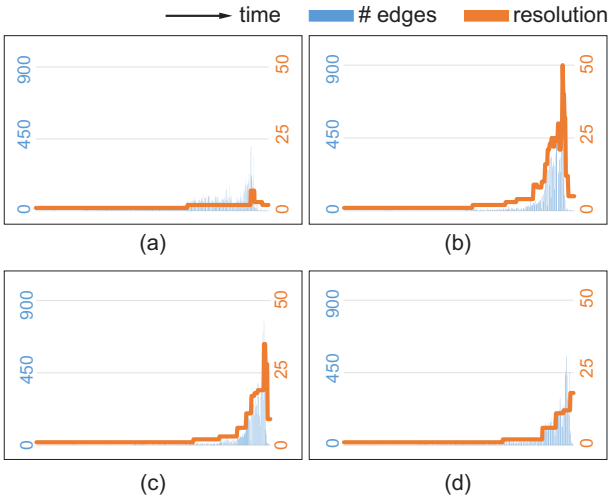

Figure 29 – Adaptive resolution and its relation with the edge distribution for the Enron network. (a) $w_{size} = 100$ and $FF = 0.9$ (921 timestamps). (b) $w_{size} = 50$ and $FF = 0.99$ (357 timestamps). (c) $w_{size} = 100$ and $FF = 0.99$ (448 timestamps). (d) $w_{size} = 200$ and $FF = 0.99$ (579 timestamps).

Figure 30 shows an approximation of the same time interval (near Dec. 12^{th}, 1999 to near May 31^{th}, 2000)[1] and the same group of nodes in three distinct layouts obtained by adopting $w_{size} = 100$ and three different Fading Factor values ($FF = 0.9$, $FF = 0.99$ and $FF = 0.99999$). The first layout, with $FF = 0.9$, maintained the original resolution during the whole interval. By doing so, each timestamp refers to a 1-day interval and so it was possible to identify days without edges. Such days are usually weekends and holidays, such as the highlighted weekend May $28^{th} - 29^{th}$, 2000 and holiday May 30^{th}, 2000 (Memorial Day). By adopting $FF = 0.99$, one can see that the weekend/holiday pattern is lost due to the aggregation of days in a single timestamp. Another pattern, however, is revealed: it is easier to identify a node without connections, i.e., a person in the company that did not receive or send any emails in this period. By analyzing the layout with $FF = 0.99999$, it is possible to see that the node without connections from the previous layout appears in the network in the last timestamp. Moreover, one can

[1] Since the resolution scale may aggregate different days in a single timestamp, the first timestamp may also represent few days before the first day of the interval depending on the adopted resolution. In the same way, the last timestamp may also represent few days after the last day of the interval.

notice that the first and the last nodes of the layout had connections only in the first timestamps. These last two patterns are visible in all three layouts, but they are more easily perceived in the layouts with higher FF values.

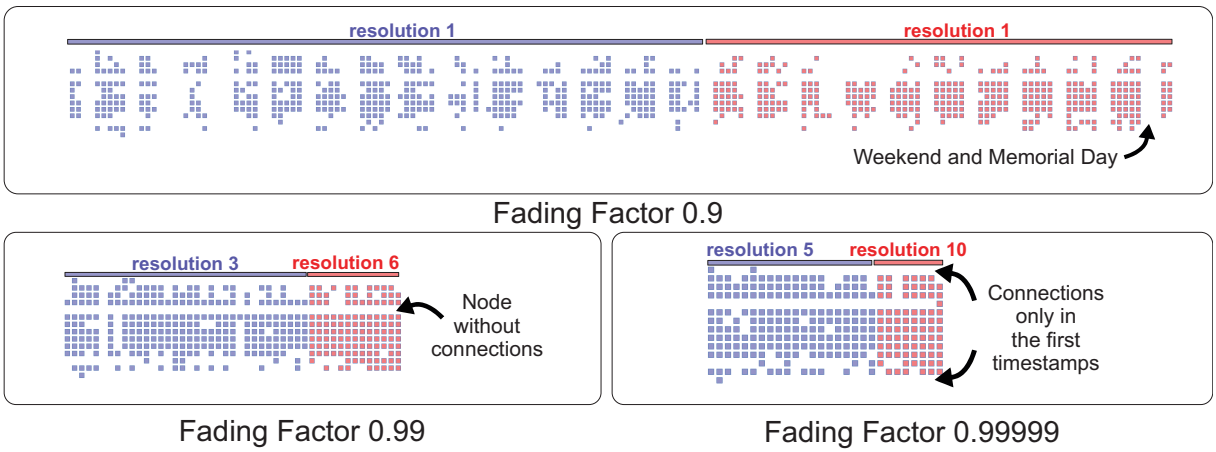

Figure 30 – Impact of different Fading Factor (FF) values on the layout ($w_{size} = 100$). The change of the node color represents a change in the resolution value. Node ordering defined by Recurrent Neighbors (LINHARES et al., 2017b).

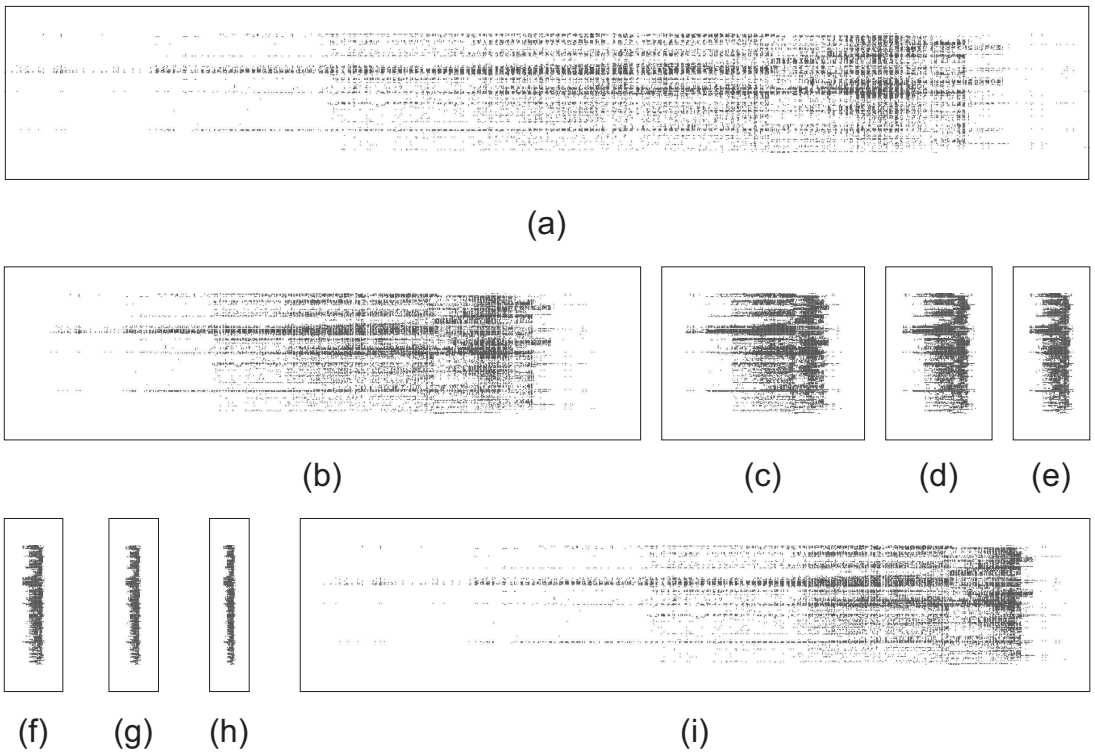

Figure 31 – TAM layouts for the Enron network considering different temporal resolutions. (a) Res. 1. (b) Res. 2. (c) Res. 7. (d) Res. 15. (e) Res. 25. (f) Res. 50. (g) Res. 75. (h) Res. 100. (i) Adaptive resolution adopting $w_{size} = 100$ and $FF = 0.9$. Node ordering defined by Recurrent Neighbors (LINHARES et al., 2017b) using Res 1.

Figure 31 shows TAM layouts for the Enron network considering different temporal resolutions. Figure 31(a-h) shows the TAM layouts for the static resolutions 1, 2, 7, 15, 25, 50, 75, and 100, respectively. Figure 31(i) shows the TAM layout generated by our adaptive resolution method ($w_{size} = 100$ and $FF = 0.9$). Resolutions 1, 2, and 7 (Figure 31(a-c)) are shown because they represent the lower (and original), the average and the higher resolution values adopted by our method for this network. The other static temporal resolutions (Figure 31(c-h)) are arbitrary values.

As illustrated in Figure 32, depending on the resolution being used, more or less patterns can be identified. The layout generated by the adaptive resolution allows the identification of at least 5 patterns (Figure 32(a)): (1) weekdays, in which there are interactions among nodes, and weekends (without interactions); (2) perception of the growing level of node activity over time; (3) identification of highly active groups of nodes; (4) a time interval with a burst of node activity near the end of the network; and (5) abrupt decrease in the number of connections followed by the end of the network. The static resolution 2 (Figure 32(b)) also allows the perception of all five patterns. However, recall that this resolution represents the average value adopted by our method, which supports the claim that it chooses resolution values that are indeed suitable for the network analysis. As expected, BVC redistributed the edges along the timestamps, and so

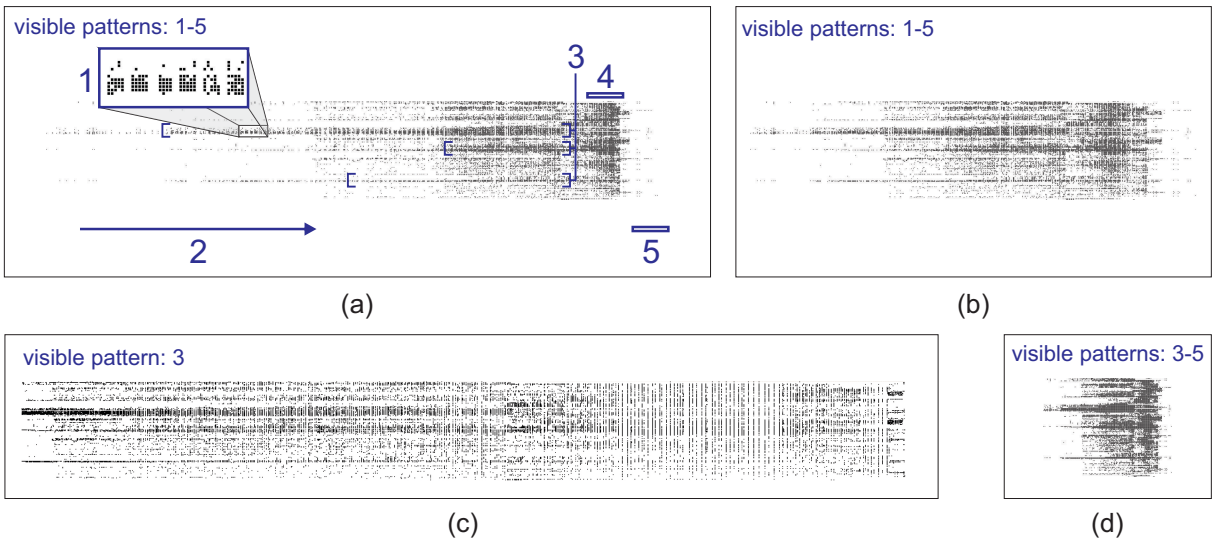

Figure 32 – TAM layouts generated by different temporal resolutions and their visible patterns for the Enron Network.Our method (921 timestamps, $w_{size} = 100$ and $FF = 0.9$). (b) Resolution 2 (673 timestamps). (c) BVC (1,345 timestamps). (d) Resolution 7 (193 timestamps). Depending on the layout, a maximum of five visual patterns can be identified: (1) weekdays (with interactions) and weekends (no interactions); (2) growing number of edges over time; (3) highly active groups of nodes; (4) burst of edges near the end of the network; and (5) abrupt decrease followed by the end of the network. Node ordering defined by Recurrent Neighbors (LINHARES et al., 2017b) using Res 1.

these temporal patterns (all but pattern 3) are lost (Figure 32(c)). By using resolution 7 (Figure 32(d)), patterns 1 and 2 are lost. Each timestamp in this resolution represents 7 days and so there is no separation of weekdays and weekends or the perception of growing node activity. One can note that the layouts with temporal resolutions above 7 (see Figure 31(d-h)) are even worse for the Enron network visual analysis.

The ideal temporal resolution depends on the network being analyzed. For the Primary School network, resolution 25 allowed the identification of several patterns (see Figure 25(b)). On the other hand, resolution 25 is not a good choice for the Enron network (Figure 31(d)). In the same way, resolution 2 would not improve Primary School analysis. Our method is capable of adapting the resolution scale according to local levels of node activity, thus enhancing the network visual analysis.

Figure 33 presents two other patterns observed in the layout with the adaptive resolution when *zooming in* the time interval with a burst of node activity showed in Figure 32(a). According to Sun et al. (SUN et al., 2007), in June 2001 occurred an important event related to the Enron accounting fraud: *"Rove divests his stocks in energy"*. In the layout, it is possible to see a decrease in the number of edges (emails) in the majority of the days in June and July involving the majority of the nodes. Such pattern may be related to this important event. The layout also shows the moment in which there is an abrupt decrease in the number of edges followed by the end of the network. Such decrease is related to the event *"Lay [Enron CEO] implicated in plot to inflate profits and hide losses"* (SUN et al., 2007), which happened in Feb 4^{th}, 2002. After the decrease of connections, the resolution of the layout was changed from 7 to 3, reflecting the new level of node activity. Temporal patterns such as these are probably lost when using BVC because of its edge redistribution (Figure 34(b)). Our method (Figure 34(c)), on the other hand, provides a distribution similar to those from the static resolution approaches

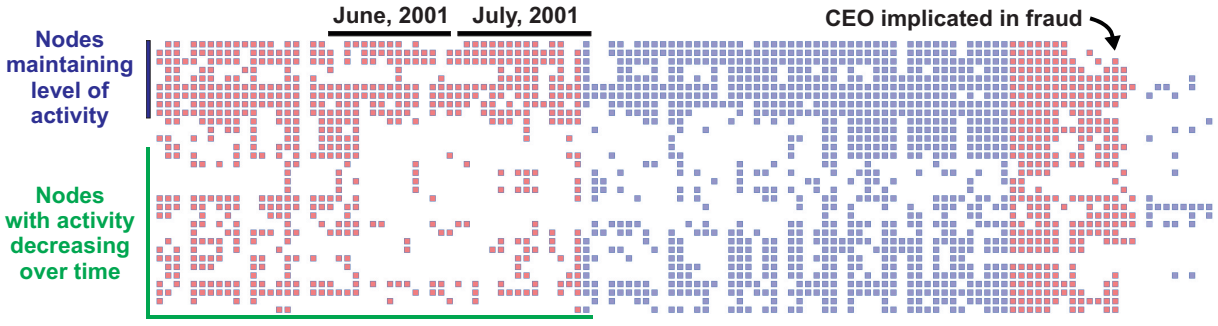

Figure 33 – TAM layout with the adaptive resolution ($w_{size} = 100$ and $FF = 0.9$) showing a portion of the Enron network. Two patterns are visible: (i) a decrease in the number of connections in June and July 2001; and (ii) an abrupt decrease in the number of connections followed by the end of the network. The change of the node color represents a change in the resolution value. Node ordering defined by Recurrent Neighbors (LINHARES et al., 2017b).

(Figure 34(a,d)), and thus is capable of highlighting such temporal patterns.

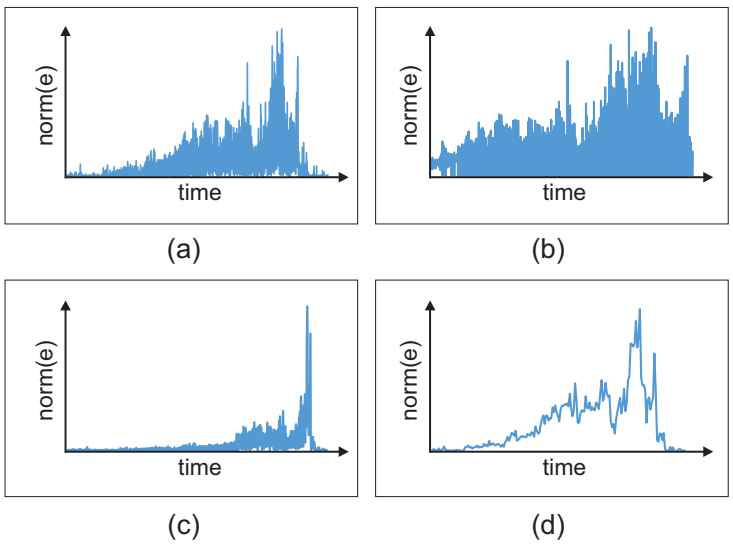

Figure 34 – Spread of edges over time according to different approaches for the Enron network. (a) Original resolution (1,346 timestamps). (b) BVC (1,345 timestamps). (c) Our method (921 timestamps, $w_{size} = 100$ and $FF = 0.9$). (d) Res. 25 (193 timestamps). While BVC changes the edge distribution because of its histogram equalization procedure, our method provides a distribution similar to those from static approaches. "norm(e)" refers to the normalization of the number of edges to values between 0 and 1.

4.3 Final Considerations

We proposed in this chapter an automatic and adaptive temporal resolution recommendation method for streaming networks. Without it, when handling temporal networks one should test different resolution scales until an adequate one is found. Besides the effort of such preliminary tests, the analyses of different networks require different temporal resolutions. For streaming networks, exploratory analysis to support the resolution choice may not be possible because edges are continuously added in non-stationary distribution. For the same reason, considering an initial set of nodes and edges to support the choice may be inefficient as well. In either case, the chosen resolution scale may not faithfully represent the distribution of future edges.

Our method automates the process of choosing a temporal resolution for streaming networks while adapts the layout according to local levels of node activity. This is possible because the choice of each new resolution scale uses only the connections from a sliding window, with old information being discounted by a forgetting mechanism. The method was applied in two real-world networks with different characteristics and the results show that the resolution values automatically adopted are indeed suitable for each network analysis.

When changing the temporal resolution, one should be aware that connections are lost to improve network comprehension. By doing so, relevant information may be lost in the process. Such characteristic exists in any sampling strategy. Our method, however, changes the current resolution according to local levels of node activity in an attempt to reduce such impairment.

Although our adaptive resolution method improves the layout by manipulating the temporal dimension, the node positioning represents an important aspect to be considered since the ordering quality may improve or impair the layout. We thus recommend that all three dimensions (node, edge, and time) should be considered in the layout construction.

Finally, since two timestamps in the layout may represent completely different time intervals, one should pay attention in the resolution scale adopted in each of them when the analysis depends on this information (e.g., to decide which node has been active for the longest time in the network). Changes in the node color, as used in the experiments, attenuate this limitation, but other solutions can be developed. In such cases, where the adaptive resolution impairs the analysis, our method remains useful as the average resolution scale computed by it represents a good choice for a static resolution scale (as occurred with resolution 25 in our Primary School analysis).

Finally, we have demonstrated our method's quality using TAM and MSV. Although our method is streaming-fashion and run for consecutive windows, these visual representations draw all network elements (nodes and/or edges) at once. This is a characteristic of these layouts and not a limitation of our method. Although they could be adapted to handle streaming scenarios by plotting consecutive windows over time, this adaptation is left as future work. Furthermore, our method does not rely on particular layouts' characteristics (e.g., length/positioning of edges or animated vs timeline layouts) and thus could be applied in different layouts as well. In animated layouts, however, the visual analysis would probably be impaired in some cases, since the number of frames devoted to high-activity periods would reduce, potentially breaking the user's mental map.

Chapter 5

Node Dimension

In the MSV layout, the positions of nodes are fixed on space and are time-invariant to maintain the mental map necessary to identify relevant patterns. Due to such constraint, an incremental node ordering focused on streaming networks would increase the visual complexity. This happens because nodes would change positions in time as they gain or lose relevance and, consequently, would pollute the layout[1].

During the streaming network analysis, the user may identify relevant regions (sub-networks) that require more detailed analysis. These sub-networks can be treated as non-streaming temporal networks and then can be stored for future or parallel analysis. In these temporal network analyses, any node ordering method can be applied. In streaming scenarios, the maximum number of edges and nodes available for analysis is limited by screen space and by the primary memory capacity, which also limits, as a consequence, the number of edges and nodes of the selected sub-networks. Existing node ordering methods, however, are not visual scalable and do not perform well when dealing with elevated number of nodes and edges (see Section 3.1.1).

This chapter presents *Community-based Node Ordering (CNO)*, a visual scalable node ordering method useful for the analysis of large and non-streaming temporal networks. CNO combines network community detection with node reordering techniques to enhance the identification of visual patterns. As CNO uses community structure information, we initially present a visual analysis method for evaluating the performance of different community detection algorithms. Both methods presented in this chapter are result of a collaboration with a (former) PhD candidate from our research group.

The presentation of the visual analysis method for evaluating different community detection algorithms (Section 5.1) is based on (LINHARES et al., 2020). Reprinted/adapted by permission from Springer Nature Customer Service Centre GmbH: Springer Nature Multimedia Tools and Applications, Visual analysis for evaluation of community detection algorithms by Claudio D. G. Linhares, Jean R. Ponciano et al © 2020. The presen-

[1] The naive *Appearance* node ordering is suitable for streaming networks if we assume that nodes cannot change position over time and plot each new node at the bottom of the MSV layout.

tation of CNO (Section 5.2) is based on (LINHARES et al., 2019b). Reprinted/adapted by permission from Elsevier © 2019.

5.1 Choosing a community detection algorithm

Although there are several network community detection methods in literature, the choice of which one to use may not be trivial since different methods may result in different communities, influencing the network analysis either by statistical measures (e.g, (YIN et al., 2015)) or visual strategies (e.g, (ROSVALL; BERGSTROM, 2010)).

Several studies in literature statistically compare the performance of different community detection algorithms (FORTUNATO; HRIC, 2016; YANG; ALGESHEIMER; TESSONE, 2016; YIN et al., 2015; ORMAN; LABATUT; CHERIFI, 2012; ORMAN; CHERIFI; LABATUT, 2011; MOTHE; MKHITARYAN; HAROUTUNIAN, 2017). Popular evaluation metrics include *Normalized Mutual Information (NMI), Adjust Rand Index, Conductance, Triangle Participation Ratio*, and others (MOTHE; MKHITARYAN; HAROUTUNIAN, 2017). In this thesis, we focus on *Precision, Recall, F-Measure, and Modularity*, which are used in (YIN et al., 2015) and briefly presented below.

- Precision: value between 0 (worst) and 1 (best) that is the ratio between the retrieved correct items and the number of retrieved items. Precision is not a good metric to evaluate community detection by itself. For instance, considering each node as a community will produce a maximum precision value (YIN et al., 2015).

- Recall: value between 0 (worst) and 1 (best) that is the ratio between the retrieved correct items and the number of corrected items. It is also not a good metric to evaluate community detection by itself. For instance, considering a single community including all nodes will produce a maximum recall value (YIN et al., 2015).

- F-Measure: value between 0 (worst) and 1 (best) that represents the harmonic mean between precision and recall. In order to have a good F-measure value, the detected communities should be close to the real ones (YIN et al., 2015). It requires a ground-truth (known communities).

- Modularity: value between -1 (worst) and 1 (best) that indicates the quality of a particular division of a network. A modularity value between 0.3 and 0.7 usually indicates a good partitioning (YIN et al., 2015). Modularity is one of the most widespread evaluation criterion used in the literature (MOTHE; MKHITARYAN; HAROUTUNIAN, 2017). It does not require a ground-truth.

Only quantitative analysis may represent a “black-box” since the user just receives a numeric output. In this way, one may not understand the relationship that exists between

this output and the network under analysis, not being possible to see how the nodes are distributed into communities, the community sizes, and other patterns. Without this comprehension, the choice of which community detection algorithm better represents a network behavior may be impaired. Visualization strategies include the user in the analysis process, making it as intuitive as possible and facilitating the network data interpretation. The user thus becomes more capable of choosing which detection method is more adequate for the context. Complementing statistical evaluation with visualization strategies improves, even more, the analysis (PERER; SHNEIDERMAN, 2008).

There are several studies in the literature that employ visualization techniques to improve the perception of structural patterns adopting communities (WANG et al., 2006; CRAMPES; PLANTIÉ, 2014; ROSVALL; BERGSTROM, 2010; TANAHASHI; MA, 2012; VEHLOW et al., 2015). However, none of them analyzed different community detection algorithms in order to choose one of them. In the following, we propose a visual analysis method to evaluate two community detection algorithms, complementing statistical evaluation and helping in the choice.

5.1.1 Visual Analysis Method

The layout to visualize the communities is constructed as follows: the nodes of a community are disposed in a horizontal line, while the communities are disposed one underneath the other (Figure 35(a)). To differentiate the algorithms, nodes from the first algorithm (Alg1) are represented by squares, while nodes from the second algorithm (Alg2) are represented by circles. This layout is used to visualize the communities returned by the two algorithms in three manners (Figure 35): (a) equivalent communities; (b) non-equivalent communities obtained by Alg1; (c) non-equivalent communities obtained by Alg2.

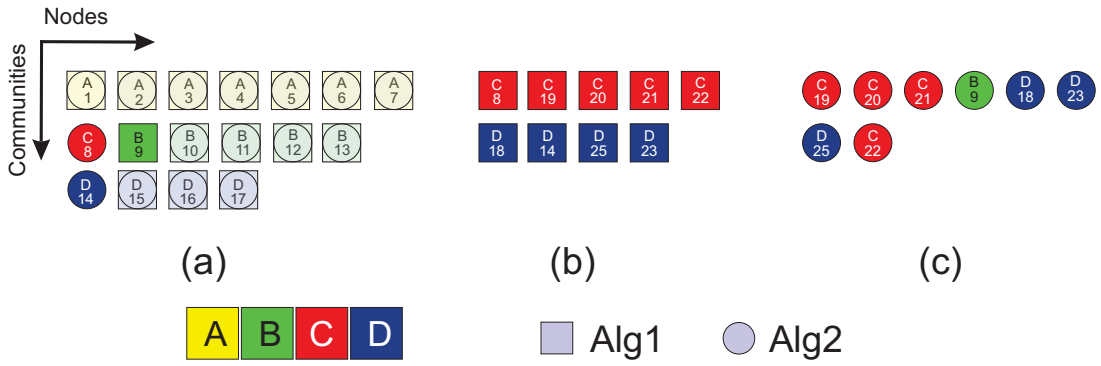

Figure 35 – Visual analysis method for evaluating two network community detection algorithms. (a) Equivalent communities; (b) Non-equivalent communities from Alg1; (c) Non-equivalent communities from Alg2. The color represents the node label, that is also written inside each node (redundant coding). Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

For the first case, we define equivalent communities as the communities returned by both algorithms that share a large number of nodes, i.e., two communities (one from Alg1 and other from Alg2) are equivalent when they share a percentage of nodes higher than a pre-defined threshold T_{Eq}. In the visualization of equivalent communities, all the nodes from both communities are shown together in the same line. However, if a node belongs to both communities, it is merged in one single representation, using both the circle and the square with transparency; otherwise it is shown separated using a square if it is only inside the community retuned by Alg1 or by a circle if it is only in the community returned by Alg2 (Figure 35(a)). In the other two cases, where non-equivalent communities are shown, the same visual strategy is employed, presenting communities detected by Alg1 (Figure 35(b)) or by Alg2 (Figure 35(c)) excluding equivalent communities.

An important element in the visualization is the representation of the node label by associating a color and a textual representation to the node (redundant coding (WARE, 2013)). This association allows to visualize the distribution of the labels inside communities. As example, in Figure 35(b) the sizes of the two communities are similar and the nodes are better distributed (they have the same labels inside each community) when compared with the communities from Figure 35(c), in which there is a disparity related to the sizes of the communities and mixed labels inside the communities.

In practice, the user may use interactive tools for zooming, panning, and to select specific nodes from one algorithm (adopting grayscale and transparency in the other nodes) and analyze how the other algorithm distributed them in its communities.

5.1.2 Case Studies

This section presents analyses involving the methods Infomap and Louvain adopting two real-world networks from different domains: a primary school and a university hospital. It is important to highlight that any two community detection algorithms could be used in the analysis. We chose Infomap and Louvain based on the discussion from Section 2.1.1. For visual evaluation, we extended DyNetVis (see Section 3.3).

For every network analysis, $T_{Eq} = 70\%$, i.e., two communities are considered as equivalent if they share at least 70% of common nodes. The impact of the T_{Eq} value in the layout is discussed later. The evaluation of the methods follows four criteria: (i) the accuracy of the returned communities, measured by *F-Measure*; (ii) the connectivity of the nodes inside a community in relation with the others outside, measured by modularity; (iii) the coherence between the amount of connections inside each community and how it interacts with other communities, which is evaluated using connection matrices; (iv) the distribution of nodes into communities and the relation of them with the node labels (when this metadata is provided by the network), evaluated by a visual analysis.

5.1.2.1 *Primary school* network

This analysis considers the *Primary School* network, described in Section 2.1.3. In this network, most of the connections occurs between students of the same class (STEHLÉ et al., 2011). Therefore, in this study we assume each class as a community to elaborate a ground truth for the evaluation of both Louvain and Infomap. As the network does not explicitly inform which teacher is designed to which class, the teachers were not considered in the ground truth. This network is the same as used in the analysis of Section 4.2.1.

Table 2 presents modularity, precision, recall and *F-Measure* for each method in the *Primary school* network. Only the analysis of precision or recall is not a satisfactory criterion on the context of community detection, but the *F-Measure* metric is useful because its value represents how similar the ground truth communities are in relation to those detected. Having obtained the higher value for *F-Measure* (0.978), Infomap obtained the best performance in this case. This value means that the communities had almost a perfect division by Infomap in relation to the classes used as ground truth. On the other hand, the modularity values (Table 2) do not suggest a great difference between both methods and thus this criterion may be ignored when choosing one of them.

Table 2 – Comparison considering modularity, precision, recall and *F-Measure* for the *Primary school* network. Higher values are better. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

Criterion	Infomap	Louvain
Modularity	0.663	0.680
Precision	0.958	0.649
Recall	1.0	0.995
F-Measure	**0.978**	0.786

Figure 36 presents the connection matrices for this network. The value of each cell represents how many interactions occur between the nodes from the communities represented by the matrix indices. As all detected communities have more connections involving only their own nodes than nodes from other communities, which is visually represented by the black background in the cells of the main diagonal, both detection methods presented coherent results in 100% of the returned communities. Besides, it is possible to see that the amount of communities returned by Infomap (10 communities) is the same of the ground truth, which does not occur when using the Louvain method.

The visualization of the communities is presented in Figure 37. In Figure 37(a) it is possible to see two equivalent communities with 100% of common nodes. One can see that these communities faithfully represent classes 1A and 1B, with their students and teachers. Figure 37(b) and Figure 37(c) show the non-equivalent communities from Louvain and Infomap, respectively. The Infomap connection matrices indicated that the number of communities returned is the same of the ground truth, but it was not possible

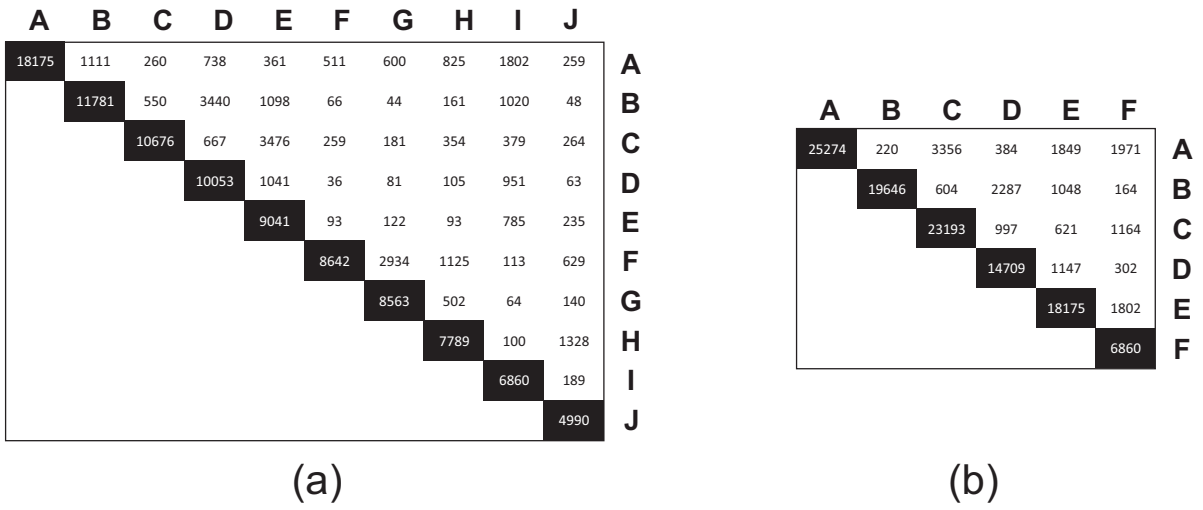

Figure 36 – Connection matrices for the *Primary school* network showing the number of interactions between the detected communities: (a) Infomap; (b) Louvain. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

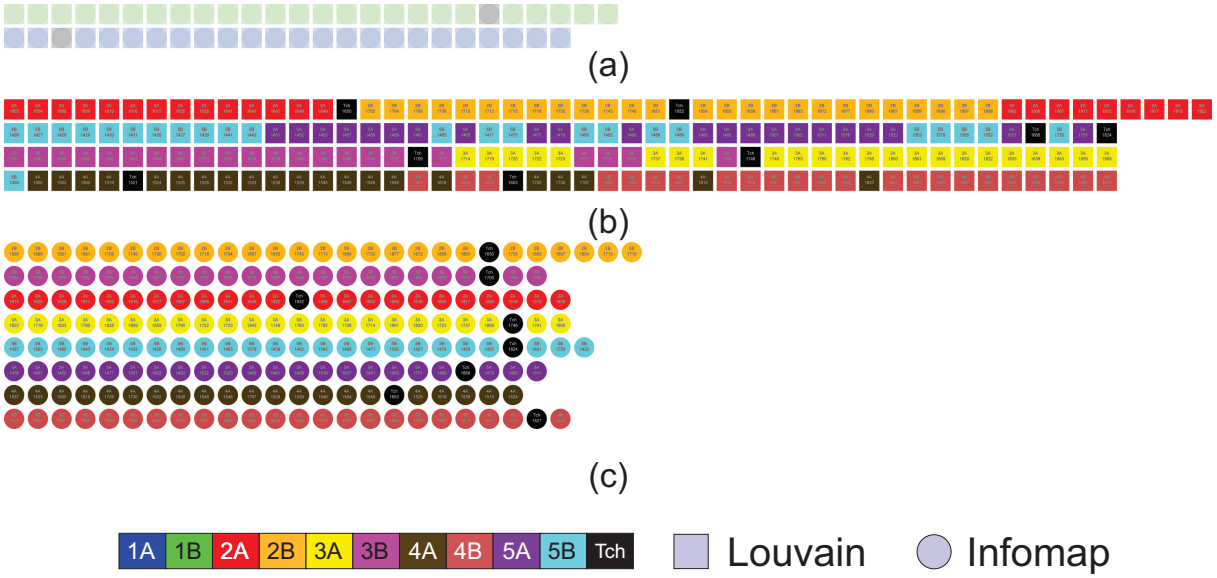

Figure 37 – Visualization of the node distribution in the communities of the *Primary school* network. (a) Equivalent communities; the non-equivalent communities obtained by (b) Louvain and (c) Infomap. The colors represent the classes and teachers. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

to see which nodes belong to which communities. In this sense, by analyzing Figure 37(a) and Figure 37(c) it is possible to see that each community faithfully represents a class and its teacher. Furthermore, although the network did not explicitly inform which teacher belongs to which class, it is also possible to identify this association by the result of the community detection of Infomap (black node in each community of the figure). With the Louvain method, this identification is not possible as there are two teachers in almost

every community.

It should be noticed that, besides the fact that Louvain returned less communities than Infomap, it merged two classes of the same grade, with exception of the first grade (1A and 1B). As students of a grade communicate more between themselves than with students from other grades (STEHLÉ et al., 2011), the Louvain result is also coherent. However, by the clear relation between class and teacher, the communities obtained by Infomap appear to be more appropriated for this network.

Although the visual analysis supported the claim that Infomap has better detection than Louvain for this network, without the visualization it would be difficult to identify that Louvain returned communities that are divided by grade whilst Infomap divided them according to the classes together with the teacher and students in each one of them. Depending on the user task in the network analysis, one detection method may be more adequate than the other. Moreover, knowing all these information may greatly influence eventual decisions making processes.

5.1.2.2 *Hospital* network

This analysis considers the *Hospital* network, described in Section 2.1.3. In this network, there is no direct relation between the communities and the people profiles, and it is expected a high degree of interaction between individuals from different profiles. In this way, it was not possible to obtain a ground truth with the available information of the network. As consequence, we can not use the *F-Measure* metric to determinate which algorithm has the best performance in this network according to this criterion. Despite that, it is possible to analyze the modularity of the network since this metric does not depend on the label of the nodes. One can see in Table 3 the modularity values for both methods. The values suggest a Louvain superiority.

Table 3 – Comparison considering modularity for the *Hospital* network. Higher values are better. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

Criterion	Infomap	Louvain
Modularity	0.313	**0.397**

Figure 38(a) shows the connection matrix of Infomap for this network. It is possible to notice that this method obtained inconsistent results for five out of the seven detected communities, which represents 71.43% of the communities: the nodes from communities C, D, E, F, and G have more connections with nodes from community A than with nodes from their own communities. The analysis of the Louvain matrix (Figure 38(b)) shows that this method also obtained an inconsistent result for three out of the six returned communities, i.e., in 50% of the communities.

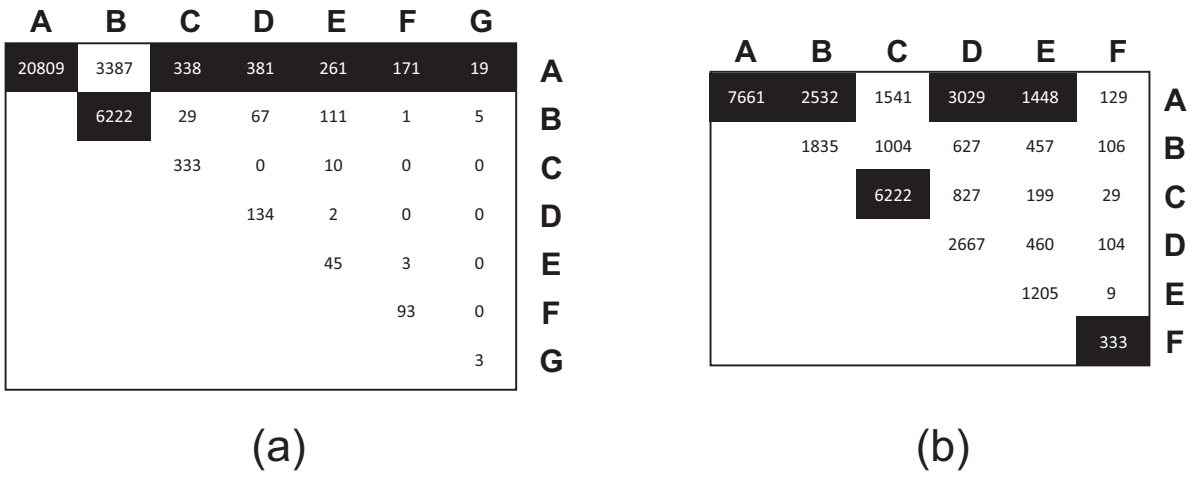

Figure 38 – Connection matrices for the *Hospital* network showing the number of interactions between the detected communities: (a) Infomap; (b) Louvain. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

The node distribution into communities in the *Hospital* network using the Infomap method (Figure 39(c)) shows the presence of one very large community and several very small ones, with only two or three elements. This situation can explain why five of the seven communities returned by Infomap have more connections with other communities than to themselves (Figure 38(a)). As previously discussed, the presence of very small

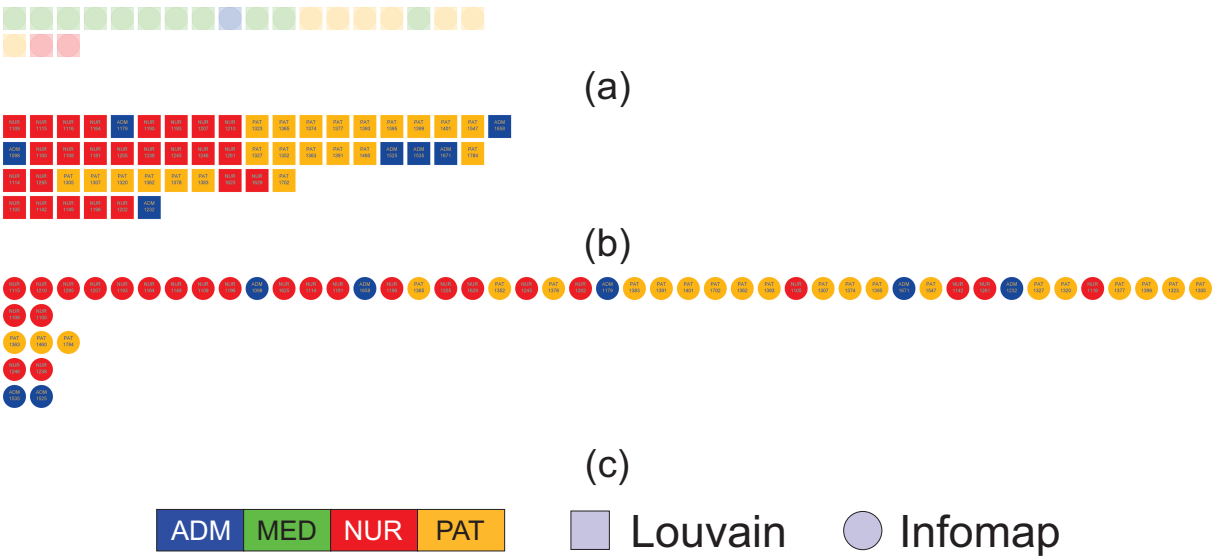

Figure 39 – Visualization of the node distribution in the communities of the *Hospital* network. (a) Equivalent communities; the non-equivalent communities obtained by (b) Louvain and (c) Infomap. The colors represent the profiles. Reprinted from (LINHARES et al., 2020) ©2020 Springer Nature.

communities is not ideal. It is possible to notice, visually comparing the node distribution of Infomap and Louvain (Figure 39(b) and Figure 39(c)), that the biggest Infomap community was divided by Louvain in small ones. Despite being expected the "spread" of profiles in communities, this situation does not occur with the MED profile in none of the detection algorithms, as can be seen in the equivalent communities (Figure 39(a)). In fact, both algorithms returned the exact same community with medical doctors. This pattern can be justified by the fact that the network represents a university hospital, suggesting that the physicians are in constant contact with the interns, thus grouping them in the same community. Besides, Louvain has shown a more coherent result (Figure 39(b)) as its community sizes are less discrepant and the communities best reflect the interaction between nurse and patient, an expected pattern in this network (VANHEMS et al., 2013).

5.1.2.3 Discussion

The analyzed networks have different behaviors and varied number of nodes and edges, which contributes to evaluate the community detection algorithms in different scenarios (Table 4). The network analysis using both community detection methods allows to perceive that Infomap returned more communities than Louvain in both cases.

Table 4 – General information of each analyzed network. Adapted from (LINHARES et al., 2020) ©2020 Springer Nature.

Network	#Nodes	#Edges	#Labels	#Commun. Infomap	#Commun. Louvain
Primary school	242	125.773	11	10	6
Hospital	75	32.424	4	7	6

The value of the threshold T_{Eq} influences the layout construction and, consequently, the visual analysis and decision making. The number of equivalent communities increases as the T_{Eq} value decreases. This is an expected behavior since all equivalent communities when adopting $T_{Eq} = n\%$ are also equivalent when $T_{Eq} = m\%$ for any $m \leq n$. In this way, one can notice that the number of equivalent communities is maximum when $T_{Eq} = 0\%$. In summary, using a small T_{Eq} value will consider as equivalent even those communities with too few common nodes, which may not represent a real equivalence. On the other hand, high T_{Eq} values will greatly restrict the occurrence of equivalent communities, which may impair analysis in real-world networks. In practice, the user may choose the T_{Eq} value through exploratory analyses.

Our evaluation demonstrated the importance of the visual analysis, since the quantitative evaluation by itself represents a "black-box" and so it can be difficult for the user to decide based only on the numeric results of a network. The inclusion of the user in this process through the visualization is important as it facilitates the decision making

of choosing the best detection algorithm for the network under analysis, enhancing the network comprehension and the perception of patterns.

The proposed visualization technique does not address visual scalability so well, perhaps with some adaptations and a further investigation this can be achieved. It is important to highlight that visual scalability still represents a major and open issue in the information visualization area (BURCH, 2017).

5.2 Community-based Node Ordering - CNO

The *Community-based Node Ordering (CNO)* is a node ordering strategy to enhance the identification of visual patterns in MSV layouts. This strategy employs community information and is based on the intuitive assumption that nodes with more connections between themselves should be close to each other. The three steps that compose CNO method are described below.

Step 1 - Community detection: Detect and categorize nodes in non-overlapping communities using any community detection algorithm in the static weighted version of the network, where weights correspond to the cumulative number of interactions over the entire observation period.

Step 2 - Inter-community reordering: Order the communities in the layout, employing any node reordering strategy by assuming a community as a single node. The idea is to reduce the edge lengths between communities and highlight connection patterns.

Step 3 - Intra-community reordering: Order the nodes in each community, employing any node reordering strategy. The idea is to refine the layout by approximating nodes with several connections among themselves inside the community.

In large networks, with thousands of nodes/edges, it is usual to observe a high number of overlapping edges, regardless of the node reordering strategy used. In this case, we can use the CNO process recursively until there is no more community to be decomposed or until a level n, informed by the user, is achieved. The idea behind this hierarchical approach is to offer degrees of granularity for the user be able to explore the data in the layout. In the first level, for instance, there could be communities that correspond to cities. In this case, the communities in the second level could correspond to neighborhoods, then streets, and so on (FORTUNATO; BARTHÉLEMY, 2007). Moreover, the user analysis of the layout may be improved through user interactions in the produced layout. Possible interactions are discussed in Section 5.2.1.3.

Figure 40 illustrates the execution of the CNO reordering for the first level. In this example, the initial sequence of nodes (A, B, ..., G) is first divided into non-overlapping

communities (Step 1, Figure 40(a) to Figure 40(b)). Next, the second and third detected communities switch places as result of the inter-community reordering (Step 2, Figure 40(b) to Figure 40(c)). Finally, as result of the Step 3 (Figure 40(c) to Figure 40(d)), the order of the nodes within each community is changed (e.g., the sequence of nodes "B,E,G" becomes "G,B,E" in the first community). Any non-overlapping detection method can be applied on Step 1. The quality of the detection algorithm directly affects the CNO results. In addition, Steps 2 and 3 are independent of each other, supporting a different reordering strategy in each step or even the same in both of them.

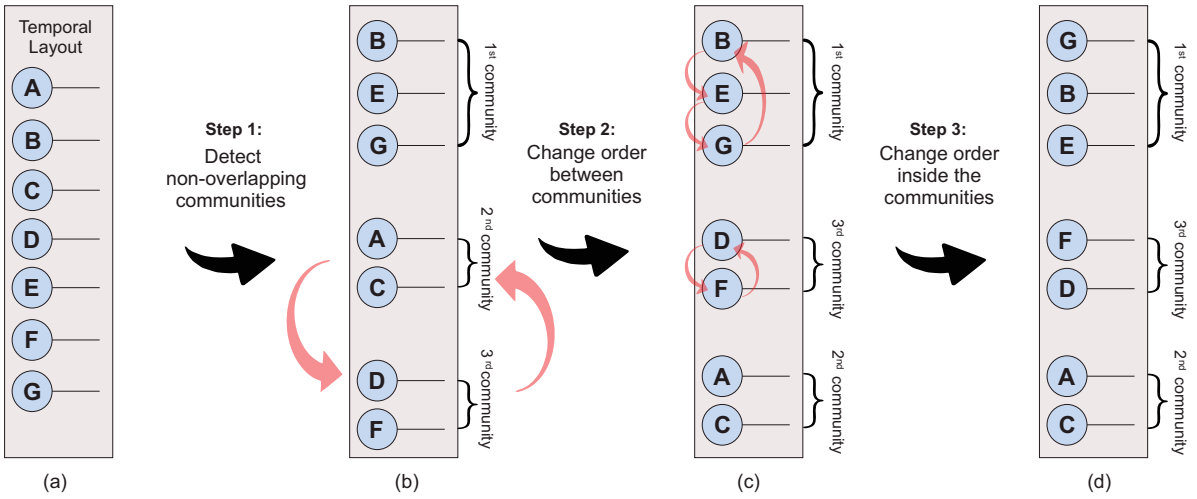

Figure 40 – Example of the *Community-based Node Ordering (CNO)* strategy considering the first level. CNO combines an algorithm to detect communities with algorithms to reorder both inter and intra communities. (a) Original network; (b) Non-overlapping communities detected in Step 1; (c) Sequence of communities after the inter-community reordering step (Step 2); (d) Final sequence of nodes returned by the intra-community reordering step (Step 3). Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

5.2.1 Quantitative and Visual Analysis

In this section, we present two case studies using real-world networks to validate CNO in scenarios representing specific phenomena, in terms of layout quality and visual scalability.

5.2.1.1 Quantitative analysis

Different node ordering methods produce layouts with different levels of visual clutter (see Section 3.1.1). To quantify the level of visual clutter and thus compare different node ordering strategies, we consider three measures suitable for the MSV layout: number of

overlapping edges (# **Overlapping Edges**); average edge length (**Avg Edge Length**); and number of intersections (# **Intersections**).

The most intuitive measure is to count the number of overlapping edges. If two edges overlap more than one time in the same time step, only one overlap is counted. This measure does not take into account that edges have different lengths and longer edges populate the layout more than shorter ones. To capture this feature, we define the length of edge (i, j) as $l_{ij} = n+1$, where n is the number of nodes in-between the connected nodes i and j and estimate the average edge length over all edges in the layout. Nevertheless, the edge length does not indicate whether a region of the image is visually dense. The number of intersections is then used to count the number of times that edges cross each other. Two edges with several intersections result on more visual clutter and thus larger and denser networks are expected to have a higher number of intersections and consequently higher visual clutter. For simplification of the comparison among different layouts, we assume the layout with the higher number of intersections as a baseline and analyze how much cleaner (less intersections) the other layouts are.

Figure 41 shows an example of the three measures computation. The three edges from timestamp 2 are positioned side-by-side to facilitate the comprehension. In practical applications using MSV layouts, the edges overlap each other.

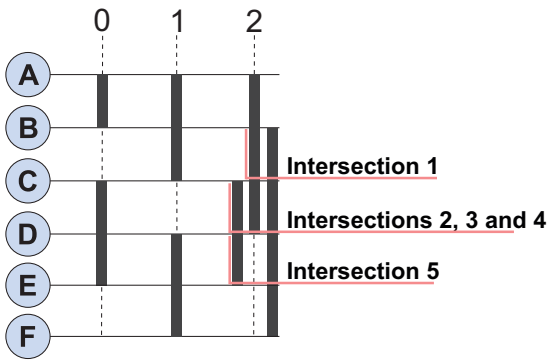

Figure 41 – Quantitative evaluation of a MSV layout. The three edges from timestamp 2 are positioned side-by-side to facilitate the comprehension. There are 3 overlapping edges in the layout; the average edge length is 2.28; and there are 5 intersections. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

5.2.1.2 Layout Decisions

To enhance pattern identification and improve the network analysis, we have made several layout decisions. The first decisions, concerning MSV layout construction (e.g., the adoption of spherical nodes and straight edges), were made based on successfully implemented approaches used in literature (ELZEN et al., 2016; LINHARES et al., 2017b). Additionally, we have made other layout decisions based on the Gestalt laws (WARE, 2013), a set of principles that describe how humans typically perceive objects in visual

representations. Based on the Gestalt's similarity law, we associate a different color to different attributes (metadata) of nodes and/or edges. In this way, it becomes easier, for example, to perceive the relationship among nodes within the same community. This color association is only possible in networks with metadata. Based on the Gestalt's proximity law, we plot an empty space between the detected communities, separating them in a way that grouped nodes are members of the same community. At last, the Gestalt's closure law requires no adaptation in the layout, since the edges disposal may visually create dense regions, that are naturally perceived as blocks of connections and concentrated activity.

5.2.1.3 Interacting with the layout

To improve visual analysis, interaction possibilities were included to allow the user to follow the Information-Seeking Mantra *"overview first, zoom and filter, then details-on-demand"* (SHNEIDERMAN, 1996), allowing analyses from global to local perspectives and vice-versa. First, the user can zoom and pan to freely navigate in the layout. Moreover, colors can be used to represent nodes' and edges' attributes, such as community membership and others. Regions of interest (e.g., edges from time x to time y or edges that connect a particular subset of nodes) can also be selected in the layout. In this case, whenever a region is selected, the elements outside it are dimmed to focus the analysis.

Besides the mentioned interaction possibilities, the user can freely navigate along the CNO levels and interactively identify the more appropriate level to perform the network analysis (Figure 42(a-c)). Moreover, the user can, at any level and depending on the task, filter in only the inter-community edges (edges that connect nodes from different communities) or only the intra-community edges (edges that connect nodes from the same community) (Figure 42(d-e)). Although there is loss of information when using the edge filtering, the objective is to highlight patterns that were not visible due to the previous level of visual clutter.

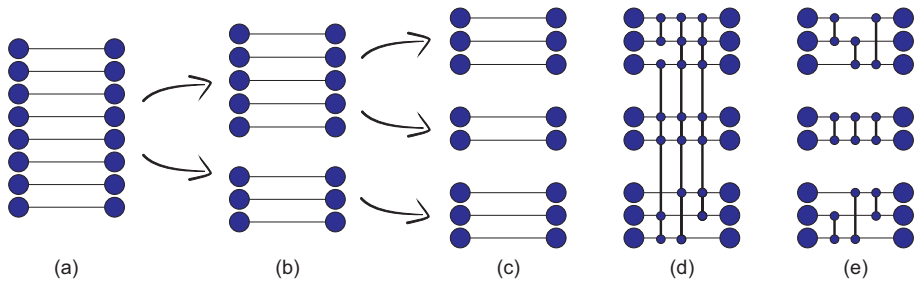

Figure 42 – Different CNO levels applied to MSV. (a-c) The CNO hierarchical approach subdivides communities at each level; (d-e) CNO allows filtering in only intra-community edges, which facilitates the analysis of specific communities. Reprinted from (PONCIANO et al., 2020) ©2020 Vilnius University Institute of Data Science and Digital Technologies.

5.2.1.4 Case Studies

We evaluate CNO using Infomap and Louvain in Step 1, and the combination of two node reordering methods (*Degree* and *Recurrent Neighbors*) in Steps 2 and 3. We have made quantitative and visual analysis to validate the quality of the CNO layout in relation to four node reordering approaches (*Appearance*, *Lexicographic*, *Degree* and *Recurrent Neighbors*). For details about these node reordering methods, please refer to Section 3.1.1. CNO was implemented and tested in DyNetVis (see Section 3.3).

Two real-world networks were evaluated through quantitative and visual analysis. The first one, called Hospital network, is a relatively small dataset and, for this reason, we decided to analyze it using only CNO level 1. This is the same network analyzed in Section 5.1.2.2. It was chosen due to its nature, in which there is a difficulty in deducing the community structure. The second dataset, called Twitter network, is considered here as a large dataset and was chosen to evaluate the CNO hierarchical approach.

Hospital Network

This analysis considers the *Hospital* network, described in Section 2.1.3. For convenience, we adopt each time as a 3-minute interval, thus reducing the number of edges to 11,977. Several node reordering algorithms may be applied on the MSV layout. Table 5 presents the quantitative results of a comparison among several of them using the measures presented in Section 5.2.1.1. Considering the number of edges e in the *Hospital* network ($e = 11,977$), it is possible to see that all algorithms present high number of overlapping edges (over 11,200).

Table 5 – Quantitative analysis using different node reordering algorithms for the Hospital network (75 nodes and 11,977 edges). For CNO, the analysis was performed using level 1. Each CNO configuration is represented by CNO (S_1, S_2, S_3), where S_x is the method used in Step x. RN: Recurrent Neighbors. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Node Reordering Algorithm	# Overlapping Edges	Avg Edge Length	# Intersections	Less visual clutter than Appearance (%)
Appearance	11,573	22.92	499,841	—
Lexicographic	11,616	20.65	405,997	18.77
Degree	11,577	16.01	284,096	43.16
RN	11,337	12.99	209,052	58.18
CNO (Infomap, RN, RN)	11,262	12.33	198,907	60.21
CNO (Louvain, RN, RN)	11,206	14.57	226,385	54.71
CNO (Infomap, Degree, RN)	11,263	12.11	194,765	61.03
CNO (Louvain, Degree, RN)	11,346	17.25	297,730	40.44
CNO (Infomap, RN, Degree)	11,332	15.15	254,278	49.13
CNO (Louvain, RN, Degree)	11,275	15.79	246,119	50.76
CNO (Infomap, Degree, Degree)	11,332	14.97	250,662	49.85
CNO (Louvain, Degree, Degree)	11,411	18.28	314,455	37.09

The number of overlapping edges does not take into account the edge lengths, so methods with similar number of overlapping edges may result in completely different layouts due to the length of such edges. As can be seen in Table 5, *Recurrent Neighbors* presents shorter edges (12.99 on average) when compared with the naive approaches (*Appearance*, *Lexicographic* and *Degree*).

As *Appearance* generates the layout with more visual clutter in Table 5 (with more intersections than the others), we take it as baseline to quantify the improvement provided by the other methods. The naive approaches present the worst results against the majority of the other algorithms. While *Lexicographic* generates a layout with 18.77% less intersections than *Appearance*, the *Recurrent Neighbors* layout is 58.18% cleaner.

The quality of the CNO results depends on the methods being used in each step. *CNO (Infomap, RN, RN)* and *CNO (Infomap, Degree, RN)* overcome the *Recurrent Neighbors* results according to all analyzed measures. On the other hand, *CNO (Louvain, Degree, Degree)* is only 37.09% cleaner than *Appearance*, a result worse than the ones from RN and *Degree*. By fixing the methods used in Steps 2 and 3 and varying the community detection method, as showed in Table 5, one can notice that the average amount of intersections when considering *CNO (Infomap, RN, RN)* and *CNO (Louvain, RN, RN)* are smaller in relation to the other node reordering combinations.

Table 6 presents the results of a comparison between *Appearance* (baseline) and CNO. The comparison considers both intra and inter-community edge filtering, varying the community detection method and adopting *Recurrent Neighbors* as node reordering strategy. Filtering the intra-community edges means that only the intra-community edges will be shown in the layout and is represented by adding the term *Intra* in the CNO configuration (e.g., *CNO (Infomap, RN, RN) Intra*). The same goes for the inter-community filtering. Using *CNO (Infomap, RN, RN) Inter* and *CNO (Louvain, RN, RN) Inter* as an example, one can see that the average edge length when considering only inter connections are high (33.79 and 26.7, respectively), which increases the average edge length of the network when considering both inter and intra-community edges (Table 5).

Table 6 – Quantitative analysis using Appearance as a baseline and the intra and inter-community filtering for the Hospital network (75 nodes and 11,977 edges). For CNO, the analysis was performed using level 1. RN: Recurrent Neighbors; Intra: Intra-community edge filtering; Inter: Inter-community edge filtering. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Node Reordering Algorithm	# Overlapping Edges	Avg Edge Length	# Intersections	Less visual clutter than Appearance (%)
Appearance	11,573	22.92	499,841	—
CNO (Infomap, RN, RN) Intra	8,777	7.91	78,519	84.29
CNO (Infomap, RN, RN) Inter	1,776	33.79	59,036	88.19
CNO (Louvain, RN, RN) Intra	3,917	2.8	7,754	98.45
CNO (Louvain, RN, RN) Inter	5,519	26.7	179,083	64.17

To analyze if the best node reordering method according to the quantitative analysis also offers the best visual experience for the user, we also performed a visual analysis. Figure 43 shows two days of the Hospital network using different node reordering strategies: (1) *Appearance*; (2) *RN*; (3) *CNO (Infomap, RN, RN)*; (4) *CNO (Louvain, RN, RN)*; (5) *CNO (Infomap, RN, RN) Intra*; (6) *CNO (Louvain, RN, RN) Intra*. The layout generated by the *Appearance* ordering presents the highest level of visual clutter amongst all strategies. Strategies 2, 3, and 4 generate similar layouts in terms of visual clutter, however strategy 4 is slightly worse (in agreement with the results shown in Table 5).

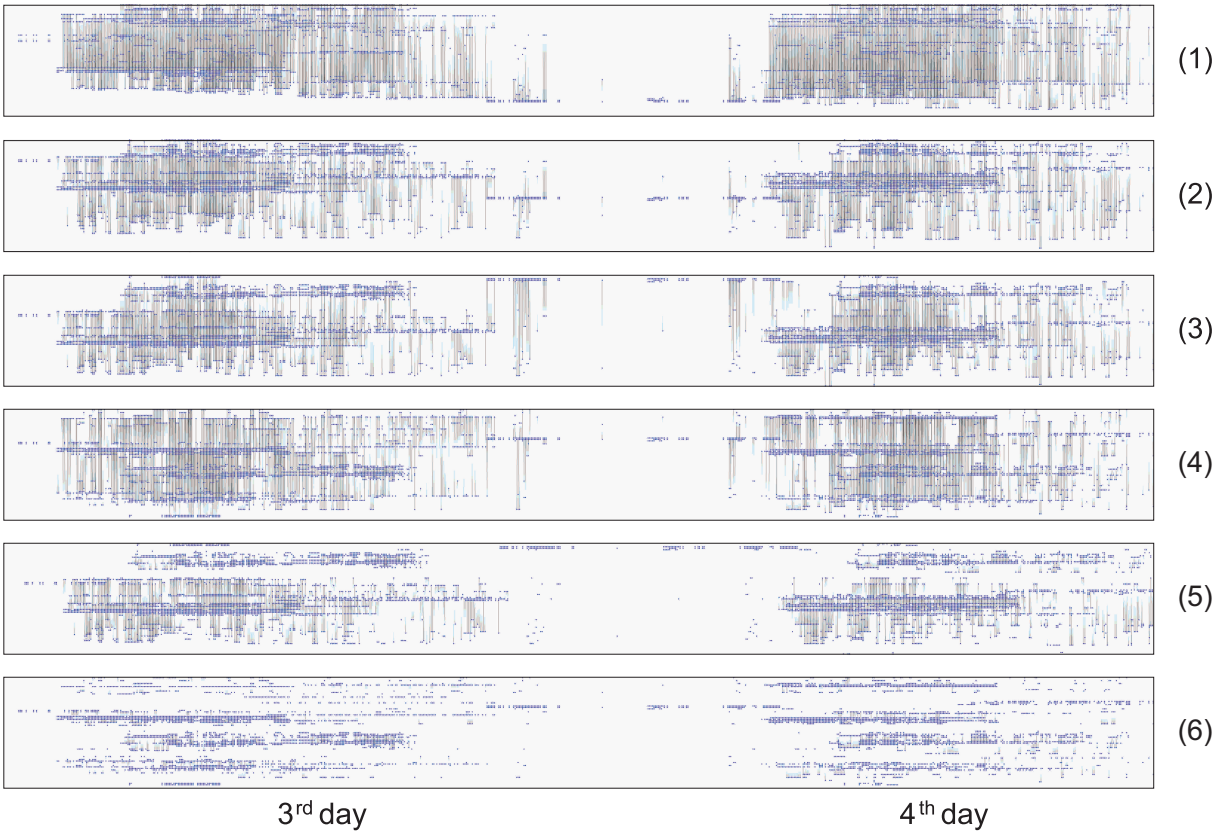

Figure 43 – An overview of the Hospital network considering two of the five days. Layouts were generated using (1) Appearance; (2) RN; (3) CNO (Infomap, RN, RN); (4) CNO (Louvain, RN, RN); (5) CNO (Infomap, RN, RN) Intra; (6) CNO (Louvain, RN, RN) Intra. Each timestamp in the layout refers to a three minute interval and considers all connections that occurred in it. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Although the analysis of inter-community edges may reveal relevant temporal patterns, we consider that such edges increase visual clutter and impair the layout, and so our focus is on the analysis of intra-community edges (Figure 43, strategies 5 and 6). The layout generated by strategy 6 is cleaner than the one generated by strategy 5, allowing easier pattern identification due to the reduced number of overlapping edges. Moreover, it allows the study of specific nodes and edges and encourages the user to further explore the

layout. Given the coherence between the quantitative evaluation and the visual analysis, the experiments on the Hospital network will use *CNO (Louvain, RN, RN) Intra* from now on.

Figure 44 illustrates the advantage of having nodes grouped by CNO. It is possible to see connections between three nodes of the NUR profile (nurses and nurses' aides). In the first and second days, only the nurse with id 1625 interacted in the network, suggesting that she/he was the only one (among the three under analysis) that was working these days. Such interactions are noticed in the layout by the presence of gray circles in particular timestamps along the days. As there are no edges associated with these nodes, the interactions do not involve any other node analyzed in the community, but others in the network. In the third day, the nurse 1629 had no interactions (she/he probably was not in the hospital this day) and both nurses 1295 and 1625 had many interactions, but not between themselves and neither simultaneously (indicating different work shifts (VANHEMS et al., 2013)). In the fourth day, the nurse 1629 also had no connections, while the others presented several interactions between themselves. This behavior is perceived by edges connecting both nodes along the day, which suggests teamwork. This behavior can also be observed in the fifth day, but with the absence of the nurse 1295 and several connections between the other two nurses. Two patterns can be observed in the presented scenario: (i) only the nurse 1625 was active in the network in all five days; (ii) the nurses 1629 and 1295 do not connect in the same day, which may be related to their days off.

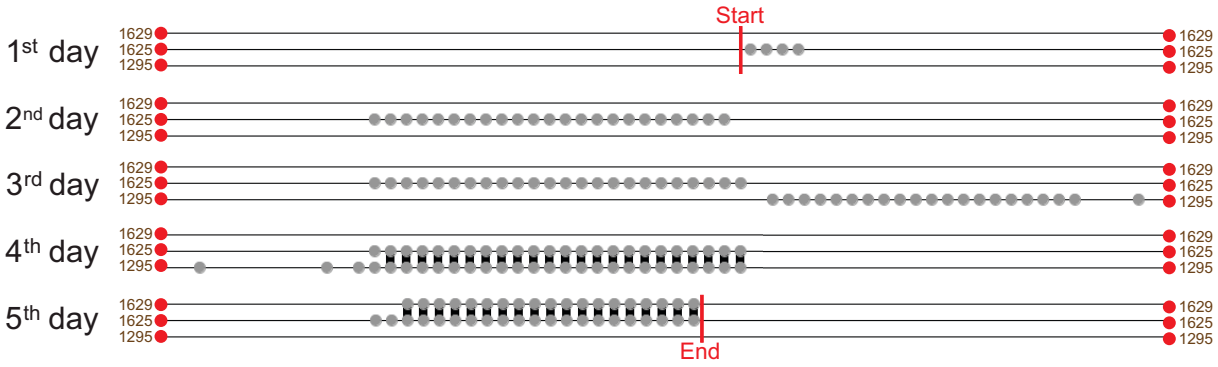

Figure 44 – Interactions between three NURs during the five days of the Hospital network. The network begins in the "Start" bar and finishes in the "End" bar. Each timestamp in the layout refers to a twenty minutes interval and considers all connections that occurred in it. Time scale per day: 12:00 a.m. to 11:59 p.m. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

It is possible to complement the layout from CNO with other visualization strategies. One possibility is using a heatmap view of node activity over time (TAM - *Temporal Activity Map*) (GHONIEM et al., 2014; LINHARES et al., 2017b), which improves the perception of certain patterns. Figure 45 shows the TAM layout with nodes from the NUR (red) and MED (green) profiles for comparing four node reordering methods (*Appearance,*

Degree, RN and *CNO (Louvain, RN, RN)*). When adopting *Appearance* or *Degree*, it is not straightforward to identify patterns in nodes from the same profile due to the distance between each other (Figure 45(a-b)). RN was able to separate the nodes according to their profiles (Figure 45(c)), facilitating, for example, the perception that all physicians join and leave the network approximately at the same time. Nevertheless, RN did not put the nodes from NUR profile as close to each other as possible, and so the identification of certain patterns involving these nodes is difficult. Using CNO (Louvain,RN,RN), the community detection algorithm grouped nurses in a community and the physicians in another and the reordering process positioned them as depicted in Figure 45(d), facilitating pattern identification. Besides the well-defined physicians' work shift, also observed with RN (Figure 45(c)), the position of nurses in the CNO layout facilitates the perception that some nurses leave approximately at the same time other nurses join the network (different work shifts (VANHEMS et al., 2013)).

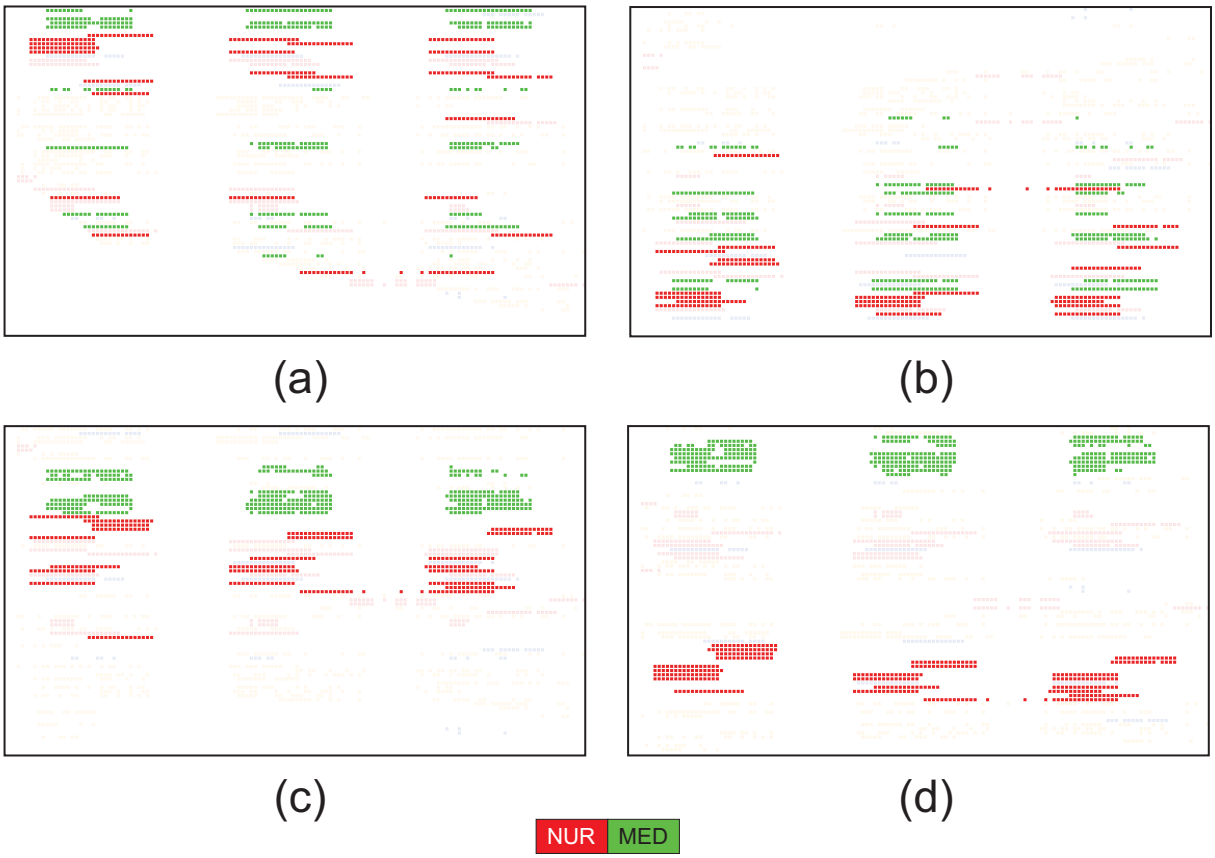

Figure 45 – Visualization of different layouts using Temporal Activity Map (TAM), for the Hospital network, with focus on profiles NUR (red) and MED (green) during three of the five days: (a) Appearance; (b) Degree; (c) Recurrent Neighbors; (d) CNO (Louvain, RN, RN). Each timestamp in the layout refers to a 30 minute interval and considers all connections that occurred in it. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Table 7 – Quantitative analysis using different node reordering algorithms for the Twitter network (50,461 nodes and 98,416 edges). For CNO, when there is no indication of which edge filtering is used, all edges are considered in the analysis. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Node Reordering Algorithm	CNO Level	# Overlapping Edges	Avg Edge Length	# Intersections	Less visual clutter than Appearance (%)
Appearance	—	98,401	13,809.97	137,780,128,036	—
Lexicographic	—	98,415	15,649.44	104,044,404,629	24.49
Degree	—	98,416	15,072.14	112,358,098,034	18.45
Recurrent Neighbors (RN)	—	98,411	8,347.97	56,738,821,369	58.82
CNO (Infomap, RN, RN)	1	96,273	4,673.5	11,782,839,610	91.45
CNO (Infomap, RN, RN) Intra	1	60,561	1,085.32	1,871,868,509	98.64
CNO (Infomap, RN, RN) Inter	1	26,822	14,235.97	8,558,588,022	93.79
CNO (Louvain, RN, RN)	1	95,877	3,131.52	7,164,444,724	94.80
CNO (Louvain, RN, RN) Intra	1	79,605	1,340.24	2,696,896,654	98.04
CNO (Louvain, RN, RN) Inter	1	14,105	13,833.36	2,486,398,635	98.20
CNO (Louvain, RN, RN)	2	95,793	3,126.08	6,849,004,216	95.03
CNO (Louvain, RN, RN) Intra	2	52,347	1,122.52	1,730,911,726	98.74
CNO (Louvain, RN, RN) Inter	2	35,531	6,654.53	4,162,333,350	96.98
CNO (Louvain, RN, RN)	3	95,794	3,127.83	6,825,292,768	95.05
CNO (Louvain, RN, RN) Intra	3	44,633	1,199.41	1,715,433,398	98.75
CNO (Louvain, RN, RN) Inter	3	40,662	5,853.81	4,202,544,529	96.95

Twitter Network

This analysis considers the *Twitter* network, described in Section 2.1.3. Table 7 presents a comparison between *Appearance*, *Lexicographic*, *Degree*, RN, and CNO. CNO was evaluated using both Louvain and Infomap methods (Step 1) and RN in Steps 2 and 3. We decided to show the results using only RN inside CNO because the layout produced by RN has almost twice fewer intersections than the one from *Lexicographic*, which was the best naive approach. For large networks, the analysis may be improved with CNO hierarchical approach. For this purpose, CNO was evaluated using three different levels in this network.

It is possible to notice that the number of overlapping edges for all naive approaches and RN include almost all edges of the Twitter network. With reference to the number of intersections, the *Appearance* method generates the layout with more visual clutter (with more intersections than the others), and so we assume it as a baseline to analyze how much cleaner are the layouts from other methods in relation to it.

CNO hierarchical approach tries to decompose the communities at each interaction. When a community is divided, some of its intra-community edges become inter-community ones in the new level. In this way, the more levels being analyzed, the less intra and the more inter edges the layout will present. With less intra edges there are less overlapping edges, as can be seen in Table 7. Comparing *CNO (Louvain, RN, RN)* in different levels, one can notice that the third level presents almost half of the intra-community edges existent in the first one (44,633 vs 79,605, respectively).

Due to the Twitter network size, the layout generated by RN alone presents a high level of visual clutter and thus cannot provide effective visual analysis (Figure 46(a)).

On the other hand, the network analysis is facilitated when adopting CNO hierarchical procedure and edge filtering (Figure 46(b-d)). With CNO, communities from level 1 are decomposed in smaller communities at level 2, and so many intra-communities edges at level 1 become inter-community edges at level 2, that are hidden when further filtering is applied Figure 46(c-d). The main advantage of this approach is that the user can study particular regions of interest in the network, as, for example, specific communities (Figure 46(e)). It is also possible to compare the activity inside and across different communities. Note also that, for the Hospital network, the layouts from CNO were usually similar to the one from RN, regardless the CNO configuration (see Table 5). However, in the Twitter network, RN always perform worse than CNO for any quantitative measure and CNO configuration.

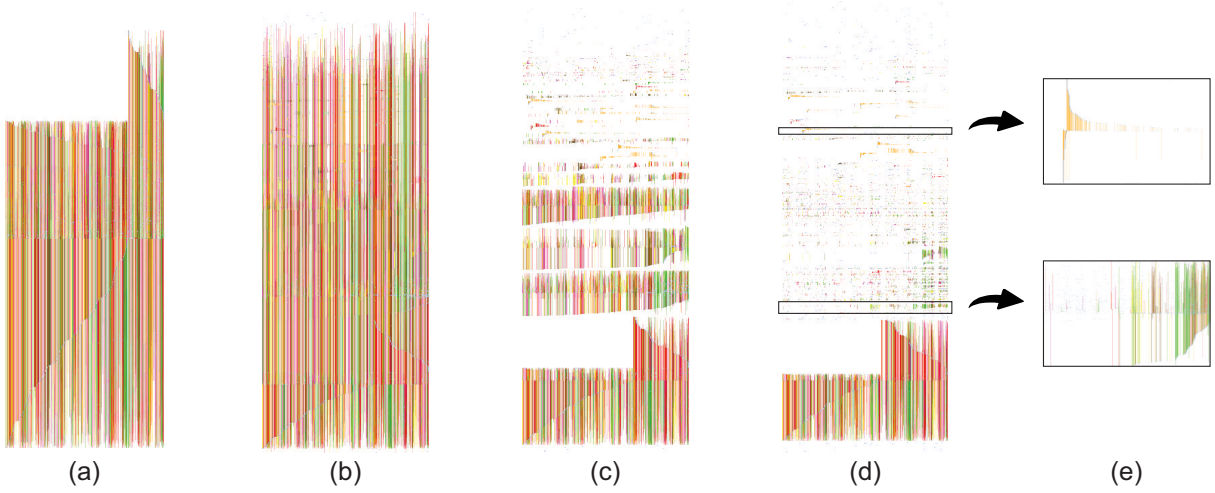

Figure 46 – An overview of the Twitter network using different node reordering strategies: (a) Recurrent Neighbors; (b) CNO (Louvain, RN, RN) level 1; (c) CNO (Louvain, RN, RN) Intra, level 1; (d) CNO (Louvain, RN, RN) Intra, level 2; (e) Examples of network communities in (d), which leads to local analysis. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

Without edge filtering, it may not be possible to perform a visual analysis due to visual clutter, especially in large networks. The Twitter network has approximately 440 edges per timestamp, which is a high number. For comparison, the Hospital network has six edges per timestamp and, although this value is many times lower, we have shown that it is hard to analyze the layout considering all edges. In order to deal with this issue, the visual analysis of Twitter network was performed using level 3 of *CNO (Louvain, RN, RN) Intra*.

The analysis of the Twitter network using level 3 resulted in a total of 3,918 non-overlapping communities, with an average size of 12.82 nodes and an average amount of intra-community edges equal to 9.49. The detected communities present different visual activity patterns:

- Communities whose members discuss about a single topic over time (Figure 47): the identification of such communities is useful because it could represent groups of individuals that share a common interest (KAPANIPATHI et al., 2014; LIM; DATTA, 2012). As an example, a commercial manager may use this information to understand the user preference and personalize product recommendation.

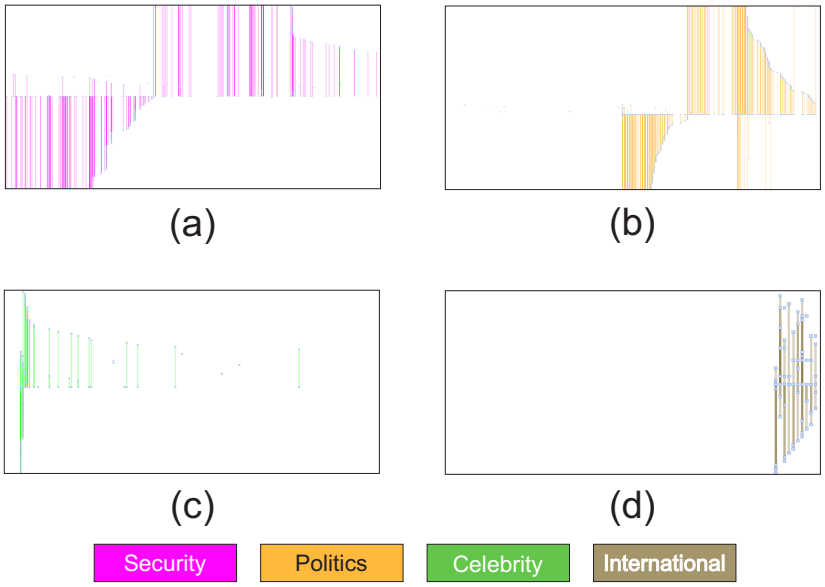

Figure 47 – Four communities from Twitter network visualized using CNO where their members discuss about a single topic over time. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

- Communities whose members discuss about different topics (topic swaps) and with variant frequency (Figure 48): the identification of topic swaps may be used to understand changes in user behaviors and seasonal products consumption (SHARMA, 2014; CARRASCOSA et al., 2013).

- Communities in which there is a spike of intra-community edges in a specific timestamp (Figure 49): the presence of a spike suggests the occurrence of a "bombshell" tweet, since it was retweeted a lot in a single timestamp (IMRAN et al., 2014). This spike represents connections between the focal node plotted by *Recurrent Neighbors* and all the others that connect to it in this timestamp (each of these nodes is visually represented by a black dot). We assume this focal node as an influential person in its community since the majority of the intra-community connections involves it. Recall the commercial manager example – in this case, the manager could invite the influencer to act in marketing campaigns.

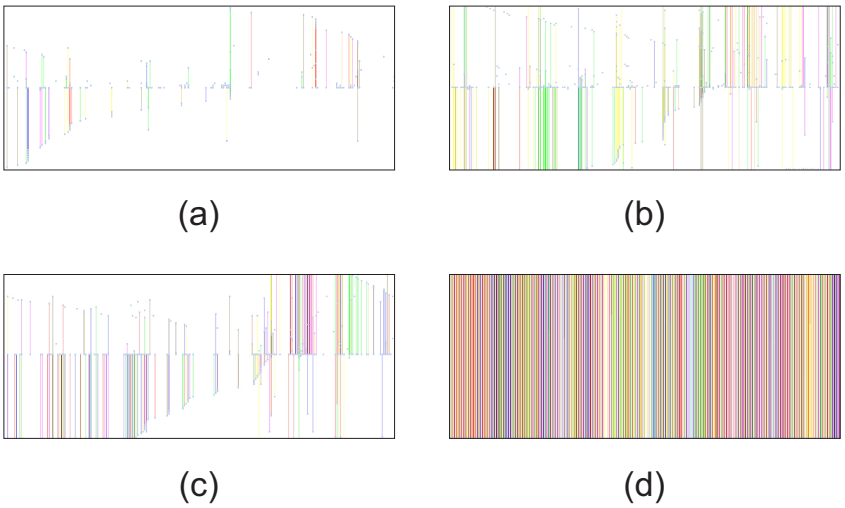

Figure 48 – Four communities from Twitter network visualized using CNO where their members discuss about different topics over time (topic swaps) and with variant frequency. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

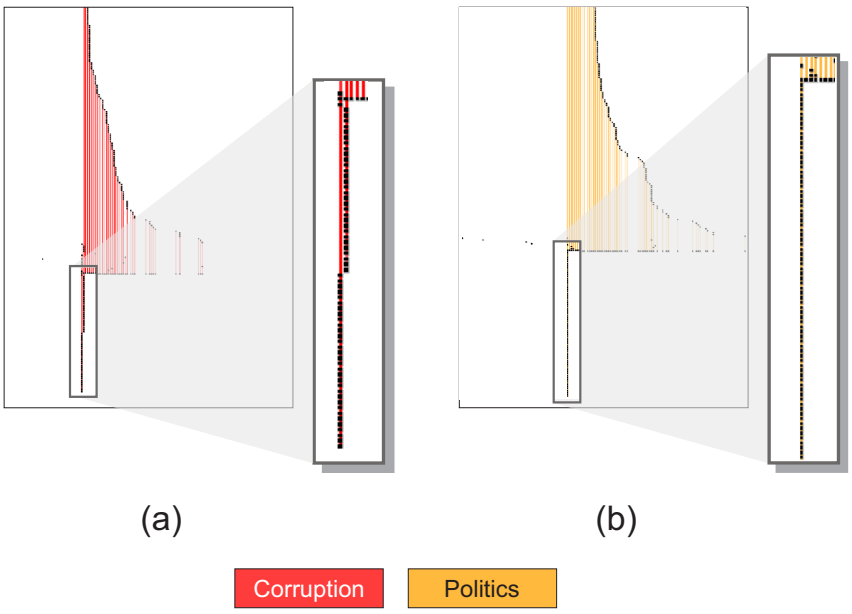

Figure 49 – Two communities from Twitter network visualized using CNO. One can see in (a) and (b) a spike in the first timestamp with intra-community edges. This behavior suggests a "bombshell" tweet since it was retweeted several times in a single timestamp. Reprinted from (LINHARES et al., 2019b) ©2019 Elsevier.

5.2.2 Limitations

As demonstrated by the experiments, CNO is a visual scalable method that improves visual analysis of large networks. Although CNO allows the user to interactively identify the appropriate level of cleaning according to the context (removing edges or breaking down in further levels), thus substantially reducing visual clutter in large networks, it has limitations when applied to very large networks. The main issue, however, is that CNO

relies on network community detection and so the method has limitations regarding the absence of community structure, as for example, in the case of purely random networks. In this scenario, CNO is expected to perform no worse than random node positioning. At last, finding the best CNO configuration is not straightforward and requires an initial exploratory analysis. The choice of the more adequate methods (to detect communities and to position nodes), the number of levels that will be considered, or which edge filtering to apply (intra- or inter-community), depends on the user task.

5.3 Final Considerations

We proposed in this chapter a novel and visual scalable node reordering method called *Community-based Node Ordering - CNO* for visual analysis of temporal networks. Our strategy improves the visual analysis of large temporal networks via a hierarchical approach. In the context of streaming networks, CNO can benefit the analysis of regions of interest (non-streaming sub-networks). We demonstrated the high quality of the CNO layout against existent node reordering methods. Besides, CNO provides an edge filtering mechanism, based on the detected communities, which improves the layout analysis by allowing the perception of patterns that would otherwise be difficult to see, especially in large networks.

The experiments performed considering the quantitative and visual analysis allowed the evaluation of CNO in scenarios considering both small and large networks. The results demonstrated coherence between the visual and quantitative analysis, and allowed the identification of several patterns that would be difficult to see without CNO, as for example, the "bombshells" in the Twitter network.

The CNO computational time complexity depends on the adopted community detection algorithm and the node reordering technique(s). The community detection methods employed in the experiments combined with the node reordering algorithms presented satisfactory computation time. Despite the chosen methods, CNO is flexible and allows the combination of any community detection and node reordering techniques.

Community detection methods are well established in the literature and are used in several contexts in the study of networks. However, the choice of which one to use may not be trivial. This chapter also presented a strategy for the evaluation of two community detection methods that uses visual analysis to help in such choice. Our study demonstrated the importance of the visual analysis, since the statistic evaluation by itself represents a "black-box" and so it can be difficult for the user to choose based only on the numeric results of a network. The inclusion of the user in this process through visualization techniques is important as it facilitates the decision about the best detection algorithm for the network under analysis.

Chapter 6

Edge Dimension

This chapter presents a novel edge sampling method named *Streaming Edge Sampling for Network Visualization* - SEVis. It removes the less relevant edges of a given network, thus highlighting bursts of connections, temporal patterns, and other structural properties. To the best of our knowledge, SEVis is the first edge sampling strategy that includes all of the following characteristics:

- It runs in a streaming-fashion and thus is suitable for streaming network analysis. Since any method developed for streaming networks is also applicable in non-streaming scenarios (AHMED; NEVILLE; KOMPELLA, 2013), SEVis can also be applied on (non-streaming) temporal networks.
- It does not require the size that the sampled network must have (regarding the expected number of nodes and/or edges) as input. It can be difficult for the user to previously define the ideal size, especially considering real-world streaming networks.
- It does not depend on specific layouts' characteristics and so it can be used in layouts generated by several visualization strategies. Each layout may provide a different perspective of the network data; SEVis is flexible and may be applied to a variety of layouts.
- There is no randomness in the sampling process. Every edge is either accepted or discarded in a deterministic manner.

6.1 Streaming Edge Sampling Method - SEVis

The streaming edge sampling method (SEVis) reduces the information from streaming temporal networks by discarding less relevant edges. Figure 50 shows the steps of SEVis. Initially, the method continuously receives incoming edges until a temporal sliding window (GAMA, 2010) defined by w_{size} times is completed. Then, at the end of the window, SEVis computes the k most relevant nodes so far in the network and verifies if a network

community detection is necessary. Several criteria may be used to define the relevance of a node. Examples include node structural properties such as closeness, betweenness, or other centrality measures (PEREIRA; AMO; GAMA, 2016a; PEREIRA; AMO; GAMA, 2016b), the frequency in which the node appears in the network (which may be computed by *Space-Saving* in a single pass (METWALLY; AGRAWAL; ABBADI, 2005)), and others. There are two situations in which a new community detection is necessary: at the end of the first window, and when the communities from the last performed detection are not suitable for representing the edges of the current window (outdated communities). Any non-overlapping community detection algorithm may be employed, as for example, *Louvain* (BLONDEL et al., 2008) or *Infomap* (ROSVALL; BERGSTROM, 2008). Similarly, different strategies can be applied to verify if the previously detected network communities represent the edges of the window, e.g., the use of anomaly or novelty detection procedures (RANSHOUS et al., 2015). Only those edges whose nodes belong to the same community (intra-community edges) are kept. From this new set of edges, any edge that has at least one of its nodes in the set of the k most relevant nodes so far in the network is accepted. The procedure is repeated for each subsequent window. Algorithm 1 presents SEVis with more details.

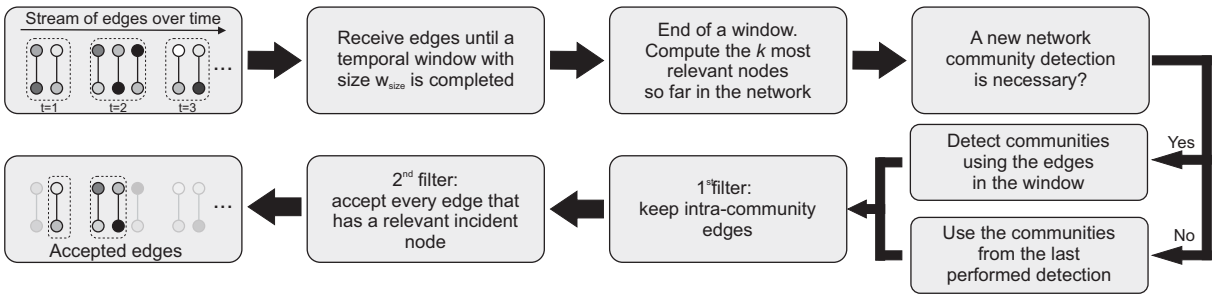

Figure 50 – SEVis workflow.

SEVis computational time complexity depends mainly on the algorithms used to detect communities, to verify whether the communities represent the current state of the network, and to compute the node relevance. All these verifications are executed only at the end of a window. For example, *Louvain* and *Infomap* are $\mathcal{O}(m)$, where m is the number of edges (FORTUNATO, 2010). Besides, community detection is performed only when needed. In the same way, *Space-Saving* is a single-pass method highly efficient for streaming tasks (SARMENTO et al., 2016). Although the performance of the methods chosen for these steps affects the SEVis computational complexity, it is relatively low under this configuration and thus the method is suitable for streaming scenarios.

Algorithm 1 SEVis edge sampling

```
1: procedure SEVIS
2:     communities ← []
3:     windowEdges ← []
4:     activeNodes ← []
5:     relevantNodes ← []
6:     for each incoming edge e(n_1, n_2, t_i) do
7:         if (t_i > 0) AND (t_i mod w_size == 0) AND windowEdges is not empty then
8:             relevantNodes ← compute the k most relevant nodes among activeNodes
9:             if (t_i == w_size) OR (communities is outdated) then
10:                communities ← detect communities in windowEdges
11:            for each edge d in windowEdges do
12:                for each community com in communities do
13:                    if d is an intra-community edge on com AND
14:                    at least one of the d's nodes is in relevantNodes then
15:                        accept d
16:            windowEdges ← []
17:        windowEdges.append(e)
18:        activeNodes.append(n_1)
19:        activeNodes.append(n_2)
```

6.2 Case Studies

This section presents case studies to evaluate the performance of SEVis in different scenarios. We analyzed randomly generated networks and two real-world networks considering both quantitative and visual evaluation. For the quantitative analysis, we initially compared SEVis against three standard sampling methods: random sampling (every edge has 50% chance of being accepted), EOD (ZHAO et al., 2018) and PIES (AHMED; NEVILLE; KOMPELLA, 2013) applied on each window (hereafter named Partial-PIES). Thereafter, we evaluate SEVis performance for varying network densities (regarding $|V|$ and $|E|$). SEVis is designed for streaming networks, so this evaluation aims to analyze its performance when dealing with sparse/dense streaming networks according to the number of edges per time. All experiments were performed using DyNetVis (see Section 3.3).

6.2.1 SEVis configuration

For community detection, we chose *Louvain* (BLONDEL et al., 2008) due to its superior performance in previous comparative studies (MOTHE; MKHITARYAN; HAROUTUNIAN, 2017; FORTUNATO, 2010) and application in recent network visualization studies (LINHARES et al., 2019b) and streaming sampling tasks (SARMENTO; CORDEIRO; GAMA, 2015b). To verify whether the detected communities represent the current state of the network, we analyze potential changes in the edge distribution. If a change is detected, we assume that there is a novelty in the window data and so a community detection

has to be performed. To discover if the edge distribution has changed, we analyze how many edges of such window are intra-community edges according to the last performed community detection. When this ratio becomes less than t_r of the average ratio over all previous windows, the community detection is re-executed. Finally, a node is considered relevant if it is among the k most frequent nodes returned by *Space-Saving* (METWALLY; AGRAWAL; ABBADI, 2005), an incremental method that maintains a list of the top-k most frequent items in a stream (GAMA, 2010). *Space-Saving* is highly efficient for streaming sampling tasks (SARMENTO et al., 2016).

6.2.2 Quantitative and visual analysis

The visual analysis is a qualitative assessment of the decrease of cluttering in the layout after sampling and its impact on visual pattern identification. The quantitative evaluation uses *Kolmogorov-Smirnov (KS)* statistic (AHMED; NEVILLE; KOMPELLA, 2013; ZHAO et al., 2018) and two cluttered-related measurements: the number of edges involved in overlaps and the number of intersections that each overlapping edge has on average, both evaluated over the MSV layout after sampling (LINHARES et al., 2019b; LINHARES et al., 2019a). KS statistic is a popular measure that assesses the distance between two cumulative distribution functions. The KS distance (KS-d) is given by the maximum vertical distance between two distributions (AHMED; NEVILLE; KOMPELLA, 2013) and varies from 0 to 1. If its value is 0, then the two distributions are identical. In our experiments, each KS distance represents the distance between the distributions of edge counts before (original distribution) and after sampling. The intersection computation for a specific time is given by the count of how many parts of an edge overlap other edges. The more intersections exist between two edges, the more they contribute to visual clutter (LINHARES et al., 2019a). Figure 51 illustrates the overlap and intersection computations for a time k (see Section 5.2.1.1 for details).

A cluttered layout impairs the visual analysis and may lead to misleading perceptions, since relevant patterns may not be identified. Having only a few edges (e.g., by using a random sampling that discards 90% of the edges) would lead to a layout with a low level of visual clutter. However, the KS distance would indicate the poor quality of the sampling due to the high discrepancy between both distributions. We thus adopt the computation of the KS distance along with the cluttered-related measurements.

Another relevant quantitative assessment is related to changes in the (non-stationary) edge distribution. Changes found in the original network should be as preserved as possible in the sampled network to support network characteristics preservation. The *Page-Hinkley* test (PAGE, 1954) is one of the most appropriate techniques to detect changes in streaming scenarios (SEBASTIÃO; GAMA, 2009). To evaluate SEVis under this criterion, we rely on the two-sided *Page-Hinkley* test with forgetting mechanism (PHT-FM) proposed in (SEBASTIÃO et al., 2013). With low delay time because of the forgetting

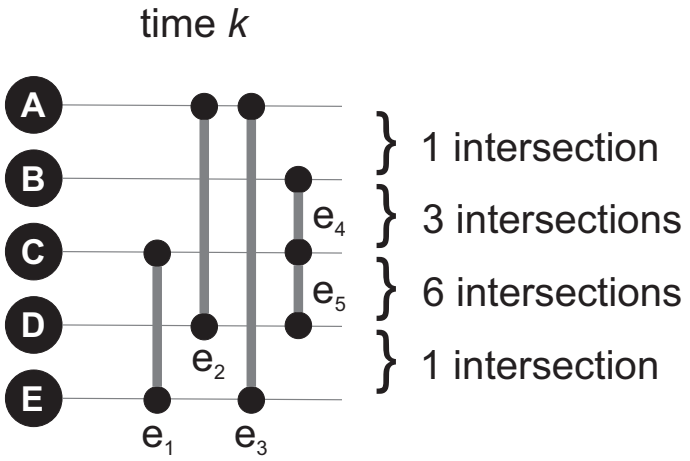

Figure 51 – Example of intersection computation. For didactic purposes, the five edges of time k (e_1, e_2, e_3, e_4, and e_5) are positioned side-by-side. The total number of intersections involving the five overlapping edges is 11 and the region between nodes C and D is responsible for 6 of them: (e_1,e_2), (e_1,e_3), (e_1,e_5), (e_2,e_3), (e_2,e_5), and (e_3,e_5).

mechanism, PHT-FM is capable of identifying both increases and decreases in the mean (two-sided) by considering the cumulated difference between the observed values and their mean until the current time. The method's results depend on the fading factor used in the forgetting mechanism (α), the magnitude of changes that are tolerated to avoid detections caused by noise (δ), and the false alarm rate (λ). A high λ value implies in few false alarms, but also increases miss detections (JRAD et al., 2017).

6.2.3 Random Modular Networks

The *Random Modular Network Generator* from Sah et al. (2014) was used to generate random networks with modular structure. We generated random modular networks with the following parameters: $|V| = 500$, average network degree equal to 6, total modules in the network equal to 10, and modularity varying from 0 to 0.9 (step 0.1). Both degree and modularity sizes follow a Poisson distribution as random modular networks with this distribution have been successfully used as benchmark to evaluate community detection algorithms, including Louvain (SAH et al., 2014).

We generated 10 random modular networks (one for each modularity value). These networks, however, do not have temporal information. To add the temporal dimension in a network, two additional steps were performed. First, we defined the desired amount of times (dt, empirically chosen as 500). Second, for each time t, $0 \leq t < dt$, we chose at random a maximum of e_{max} of network edges and associate them to t (e_{max} empirically chosen as 0.05). This procedure was repeated 10 times for each of the 10 original random networks, resulting on 100 networks with $dt = 500$ and $e_{max} = 0.05$. Table 8 shows the parameters for each evaluated method. The *Appearance* node ordering, that sorts nodes

according to the timings of first connection (LINHARES et al., 2019a), was applied in each random network to generate the MSV layout used to compute the number of overlapping edges and intersections.

Table 8 – Parameters used in the execution of each edge sampling method.

Method	Parameter values
SEVis[1]	$w_{size} = 100$ $k = 0.25 \times \|V\|$ $t_r = 0.8$
Random	50% of probability to accept an edge
EOD[2]	$\sigma = 5$ (Gaussian kernel bandwidth) F = 2 (sampling factor)
Partial-PIES	$w_{size} = 100$ node reservoir size $= 0.8 \times \|V\|$

Figure 52 shows the results for the random networks. In each plot, results correspond to averages over 10 random networks with the same set of parameters. Partial PIES presented higher KS distance (in relation to the original network) than the other methods, which may be justified by the replacements of edges in the *reservoir* (Figure 52(a)). This replacement discards mainly the older edges of the window and thus produces high discrepancies in such time intervals. The naive random sampling presented the lowest KS distance, which is expected since approximately half of the edges of each time are discarded, thus maintaining a distribution of edge counts that is similar to the original. SEVis and EOD had similar performance for modularity between 0 and 0.3. For higher modularity values, SEVis performs better than EOD, with KS distance reducing and stabilizing around 0.05. This happens because the higher the modularity, the more well-defined the community structure and, consequently, the better the community detection quality of SEVis.

Figure 52(b) shows the number of accepted edges that are involved in overlaps in the MSV layout generated by each sampling method. Although Random sampling achieved the lowest KS distance (Figure 52(a)), more than 99% of its accepted edges overlap at least one other edge in the corresponding MSV layout. Partial PIES and EOD also presented high overlapping ratios. On the other hand, the number of overlapping edges of SEVis is close to 40% for modularity values between 0 and 0.5 and never becomes higher than 80% in the evaluated scenarios, thus indicating its capability of generating cleaner layouts. Two overlapping edges that have many intersections between themselves contribute more to visual clutter than two edges that share few intersections. As it can be seen in Figure 52(c), Random sampling and Partial PIES have more intersections than

[1] In our experiments, we rely on $|V|$ to define the value of k. In real-world streaming scenarios, $|V|$ is unknown, so the value of k should be defined (or adapted) according to another criterion (e.g., first window data or domain expertise).

[2] The parameter values adopted for EOD follow those used in Ref. (ZHAO et al., 2018).

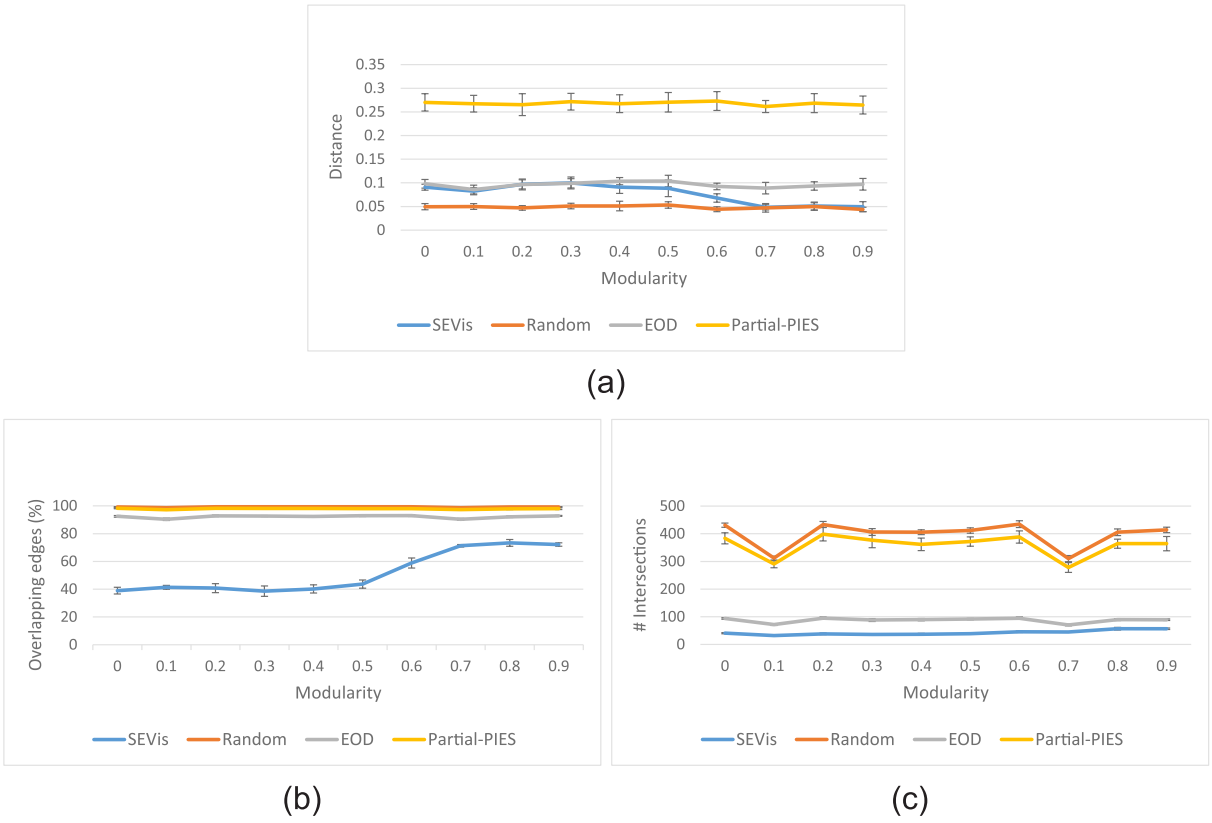

Figure 52 – Quantitative evaluation comparing SEVis, Random sampling, EOD, and Partial PIES. (a) KS statistic - distance between the distributions of edge counts before and after sampling. (b) Number of overlapping edges in the corresponding MSV layout. (c) Number of intersections in the corresponding MSV layout. The same node ordering (*Appearance*) was used in all analyses. The result obtained for each modularity value represents the average over 10 generated random networks.

EOD and SEVis, thus resulting in more cluttered MSV layouts and, consequently, in fewer visible patterns. On the other hand, SEVis is the method that generates the lowest number of intersections, supporting the claim that it produces layouts with less visual clutter and is thus more suitable for visual analyses.

To evaluate SEVis for different network densities, we generated random modular networks with 1,000 and 10,000 nodes. For each network size, we adopted average network degree equal to 50, total modules in the network equal to 10, modularity equal to 0.5, $dt = 500$, and $e_{max} \in \{0.001, 0.01, 0.05, 0.10\}$. Both degree and modularity sizes follow a Poisson distribution. Table 9 presents the average number of edges per time calculated over 10 random modular networks.

Figure 53 shows SEVis performance evaluated using the KS distance before and after SEVis sampling for each generated network. In all cases, the distance between the original and SEVis distributions is less than 0.15, indicating good results achieved by SEVis. Moreover, as the number of edges in the network increases, the KS distance decreases,

Table 9 – Average number of edges per time calculated over 10 random modular networks for each e_{max} and number of nodes. Adopted parameters: $|V| = 1{,}000$ or 10,000, average network degree equal to 50, total modules in the network equal to 10, modularity equal to 0.5, and $dt = 500$. Both degree and modularity sizes follow a Poisson distribution.

Number of nodes ($\|V\|$)	$e_{max} = 0.001$	$e_{max} = 0.01$	$e_{max} = 0.05$	$e_{max} = 0.10$
1,000	12.17	124.61	625.66	1,257.56
10,000	123.32	1,258.29	6,178.68	12,503.5

likely because of the better community detection performance. More representative communities lead to more relevant intra-community edges, which are used in the sampling process. For 10,000 nodes, there are 123 edges on average per time when adopting e_{max} = 0.001 and the corresponding KS distance is equal to 0.13 ± 0.02. On the other hand, there are 12,503 edges on average per time when considering $e_{max} = 0.10$. Even with this high number of edges, the KS distance remains low (0.019 ± 0.003), supporting the suitability of SEVis for the study of large networks. Furthermore, unlike EOD, that has computational time complexity $\mathcal{O}(m^2)$ (ZHAO et al., 2018), where m is the number of edges and therefore cannot be applied on large or streaming networks, SEVis is suitable and presents good results in such scenarios.

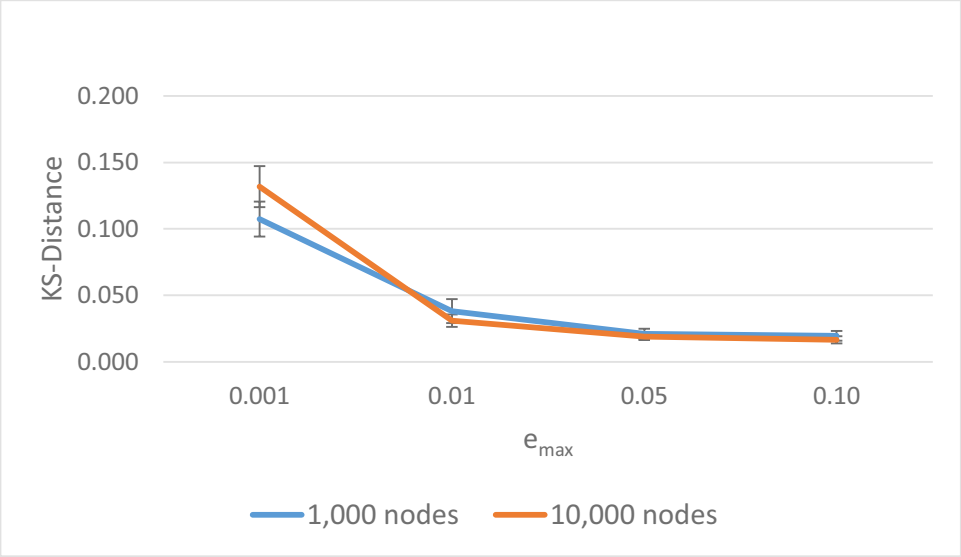

Figure 53 – Evaluating SEVis performance for different network densities. The networks were generated using the parameters in Table 9.

6.2.4 Real-world Networks

This section presents three case studies that analyze the SEVis performance in real-world networks with different sizes and densities. The goal is to visually compare the layouts generated by different sampling methods and identify which layout allows a faster

and more reliable pattern identification. Whenever the analysis considers change detection, we employ PHT-FM with $\alpha = 0.9999$, $\delta = 0.1 \times avg(e_t)$, and $\lambda = avg(e_t) + std(e_t)$, where $avg(e_t)$ is the average and $std(e_t)$ is the standard deviation of the number of edges per timestamp considering the aggregated network. It is important to note that we rely on the aggregated network to define the values of δ and λ, but in real-world streaming scenarios one should compute these values by other means, e.g., by considering only the first window.

6.2.4.1 Enron Network

This analysis considers the *Enron* network, described in Section 2.1.3. Figure 54 presents six MSV layouts generated by different edge sampling methods. The parameters are the same as those in Table 8, but with $w_{size} = 50$ for SEVis and Partial PIES. The chosen node ordering was given by CNO(Louvain, RN, RN), i.e., by CNO using *Louvain* for network community detection and *Recurrent Neighbors* for community and node ordering (for details, please refer to Section 5.2). Figure 54(a) shows the original network, without sampling. Figure 54(b) shows the layout obtained after EOD sampling. CNO(Louvain,RN,RN) inter- and intra- community edge sampling are shown in Figure 54(c-d), respectively. The layouts from Partial PIES and SEVis are shown in Figure 54(e-f), respectively.

By observing SEVis layout, one may note that SEVis allows the identification of highly active groups of nodes over time (indicated by blue arrows in Figure 54(f)) more easily than any other method, except CNO intra-community sampling (Figure 54(d)). CNO, however, requires all edges in primary memory. Not least, SEVis also allows the identification of a time interval with high activity near the end of the network. The higher activity in such interval is not identified when using CNO intra- or inter- community sampling (Figure 54(c-d)). It is partially visible, however, with Partial PIES (Figure 54(e)) and EOD (Figure 54(b)), but not as clear as with SEVis. Recall that EOD also requires all edges in primary memory. Furthermore, Partial PIES discards potentially relevant information after sampling due to the edge replacement in its *reservoir* (see Section 6.2.3). The mentioned high activity interval is related to important events that occurred at the Enron Inc. More specifically, it represents the increase and decrease of email communication following the company's CEO resignation, fraud investigation, and bankruptcy (LINHARES et al., 2017b). SEVis allows the identification of patterns related to both groups of nodes and temporal events. Contrary to CNO and EOD, it is suitable for streaming scenarios, thus being the best edge sampling among the evaluated methods.

Figure 55 shows the number of edges per timestamp in the Enron network before (original) and after SEVis, along with the timestamps in which changes were detected in both of them (orange bars). SEVis discarded 42% of the network edges. Even with such reduction, SEVis maintained relevant characteristics of the original network measured

by the KS distance (KS-d = 0.07) and PHT-FM (detection of three of the five changes originally detected (see Figure 55(b)).

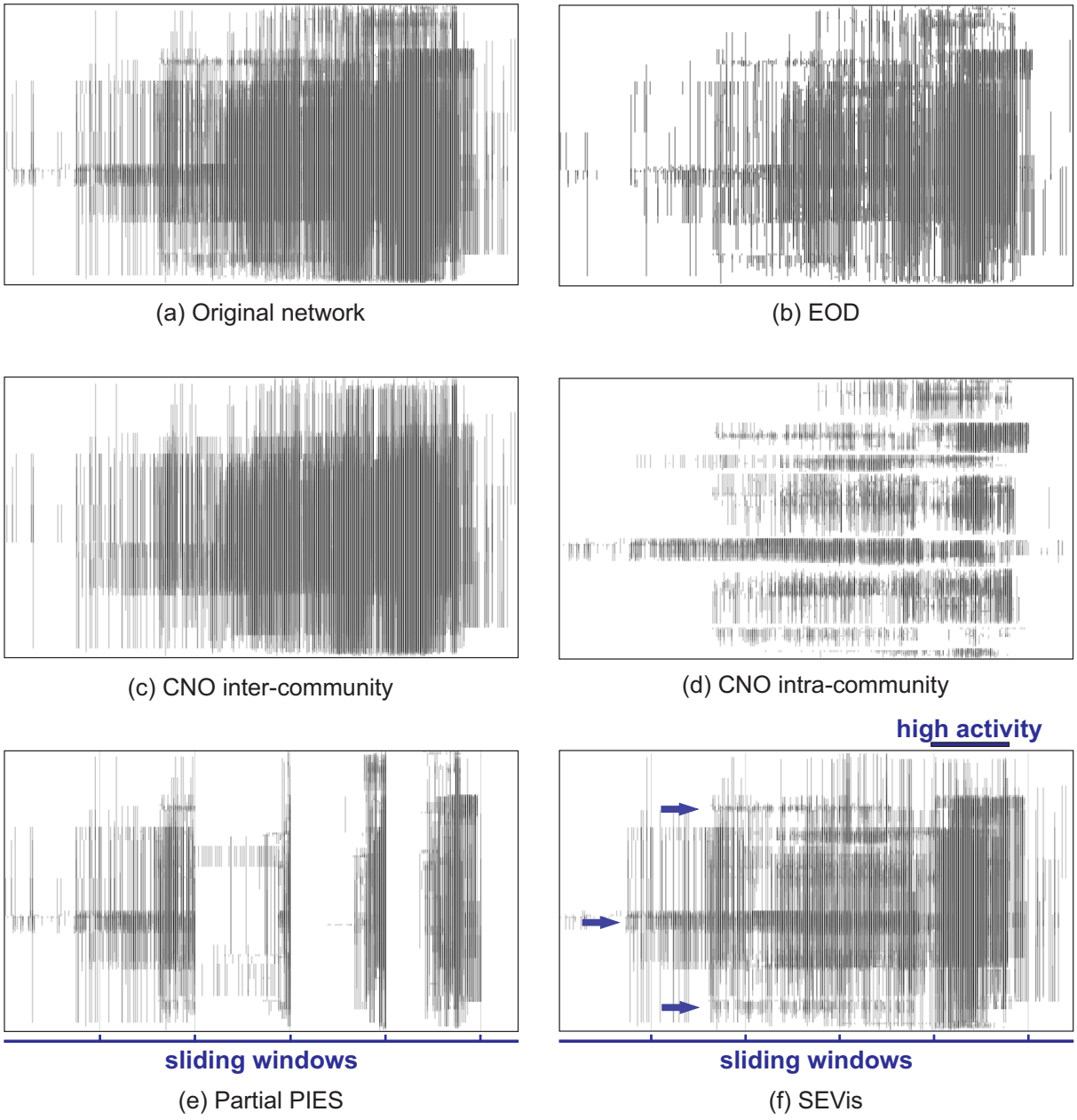

Figure 54 – Visual evaluation of different edge sampling methods for the Enron Network using MSV. Node ordering defined by CNO(Louvain,RN,RN). (a) Original network, (b) EOD, (c) CNO inter-community sampling, (d) CNO intra-community sampling, (e) Partial PIES, and (f) SEVis. EOD and CNO are not suitable for streaming scenarios, since they require all edges in primary memory. Each time in the layouts refers to a 5-day interval. Blue arrows indicate highly active groups of nodes.

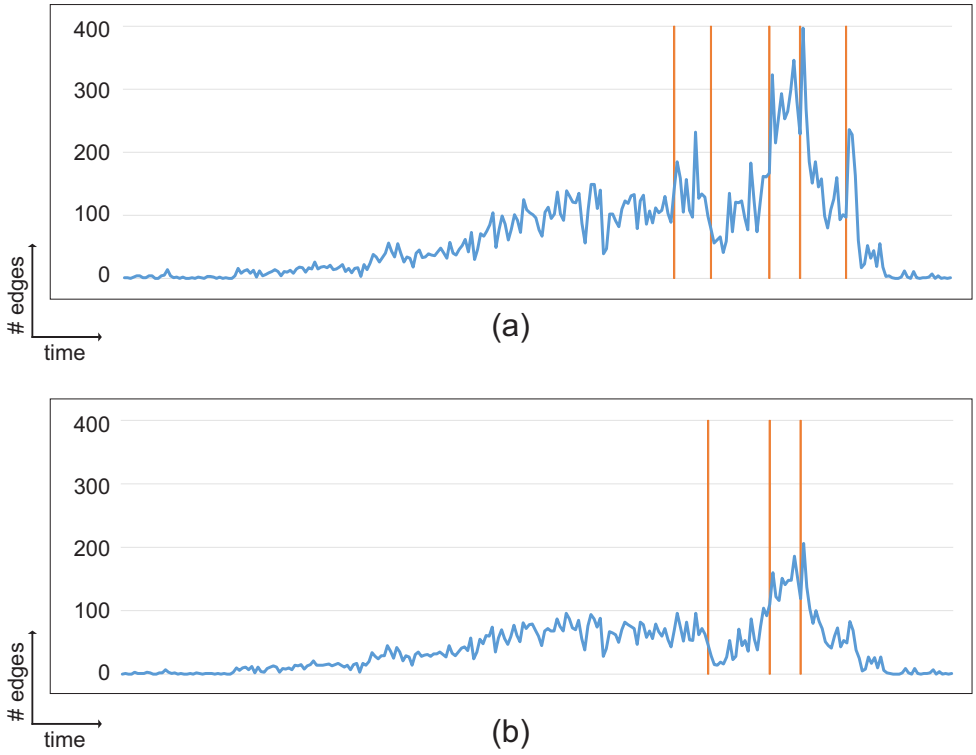

Figure 55 – Number of edges per timestamp (blue) and timestamps where changes were detected by PHT-FM for the Enron network (orange). (a) Original network. (b) Network after SEVis sampling. SEVis discarded 42% of the edges while maintaining relevant characteristics of the original network measured by the KS distance (KS-d = 0.07) and PHT-FM (detection of three of the five changes originally detected). The SEVis configuration and the temporal resolution are the same as from Figure 54.

6.2.4.2 Hospital Network

Figure 56 presents three MSV layouts for the Hospital Network (for details about this network, please refer to Section 2.1.3). Figure 56(a) shows the original network, before sampling. The layouts obtained after EOD and SEVis sampling are presented in Figure 56(b-c), respectively. Once again the parameters are the same as those in Table 8, but with $w_{size} = 50$ for SEVis, and the chosen node ordering was given by CNO(Louvain, RN, RN).

In this network, SEVis also allows identifying highly active groups of nodes. As an example, the nodes at the top of each layout represent members of the MED profile, but only SEVis highlights several interactions between them, especially in days 1 and 2 (green brackets in Figure 56(c)). Since the network represents a university hospital, it is expected a high level of interactions involving physicians and medical students (VANHEMS et al., 2013). Similarly, SEVis allows perceiving time intervals with high level of activity between specific nurses and nurses' aides in days 2 and 3 (red brackets in the figure). Not least, it is easier to identify night periods (long time intervals with few or no activity) using SEVis

because it considers as non-relevant most of the night contacts and thus discards them. In fact, only 5.9% of the interactions occur during nights in this network (VANHEMS et al., 2013).

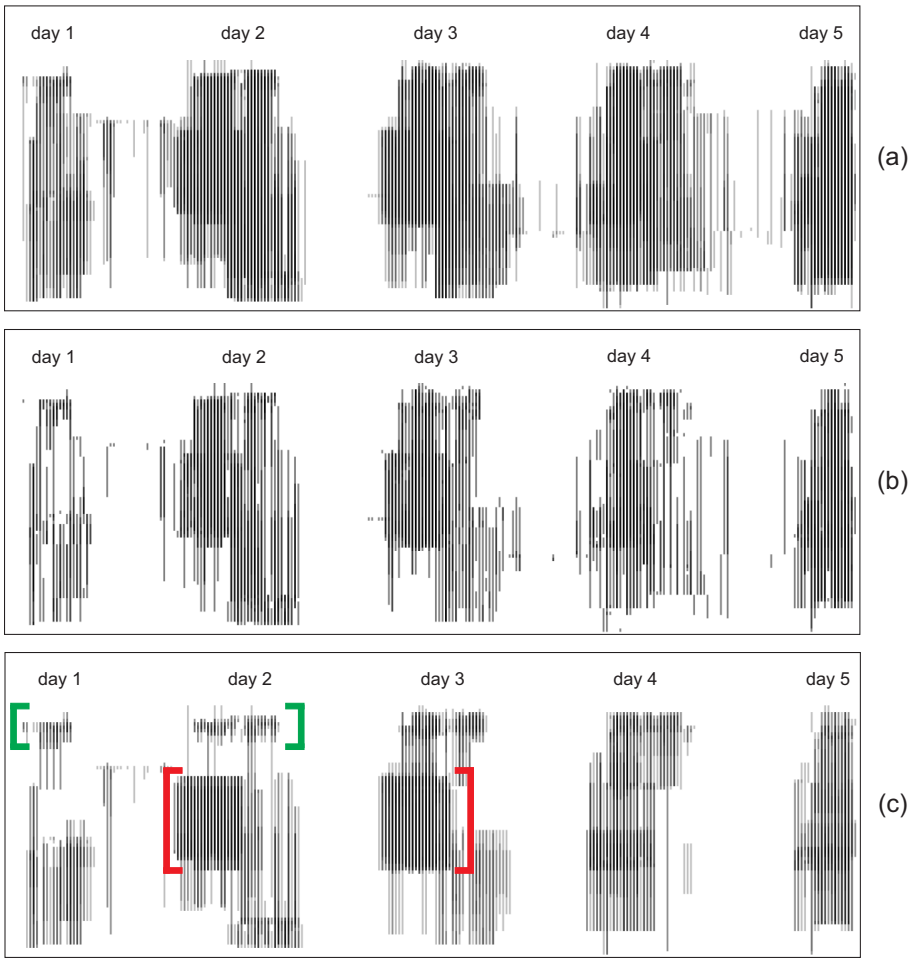

Figure 56 – Visual evaluation of different edge sampling methods for the Hospital Network using MSV. Node ordering defined by CNO(Louvain,RN,RN). (a) Original network, (b) EOD, (c) SEVis. The brackets in (c) indicate some regions of high level of activity between MED profile (green) and NUR profile (red). Each time refers to a 23-minute interval.

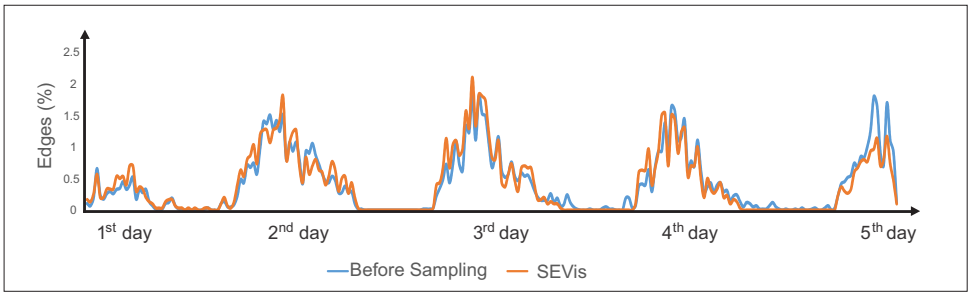

Figure 57 – SEVis distribution of edge counts for the Hospital Network. The parameters are the same as those in Table 8, but with $w_{size} = 50$ for SEVis. Each time refers to a 23-minute interval.

The Hospital Network had originally 5,896 edges (Figure 56(a)) and SEVis maintained 3,058 of them (51.8% – Figure 56(c)) while preserving the original distribution (KS-d = 0.14). Figure 57 shows the original and the SEVis distribution of edge counts for the Hospital Network. There was more discrepancy between both distributions at day 5. This may be due to the incomplete last window, i.e., the network ended in the middle of a window and consequently SEVis had only partial data to process.

As previously mentioned, this network contains time intervals characterized by a high activity level (day periods) and time intervals with few or no activity (night periods). As illustrated in Figure 58(a), PHT-FM was capable of identifying increases and decreases in the activity level in all but the first day of the original network (orange bars). Even with SEVis discarding 48.2% of the edges, this characteristic was preserved, i.e., changes were detected in all days that presented changes in the original network (Figure 58(b)).

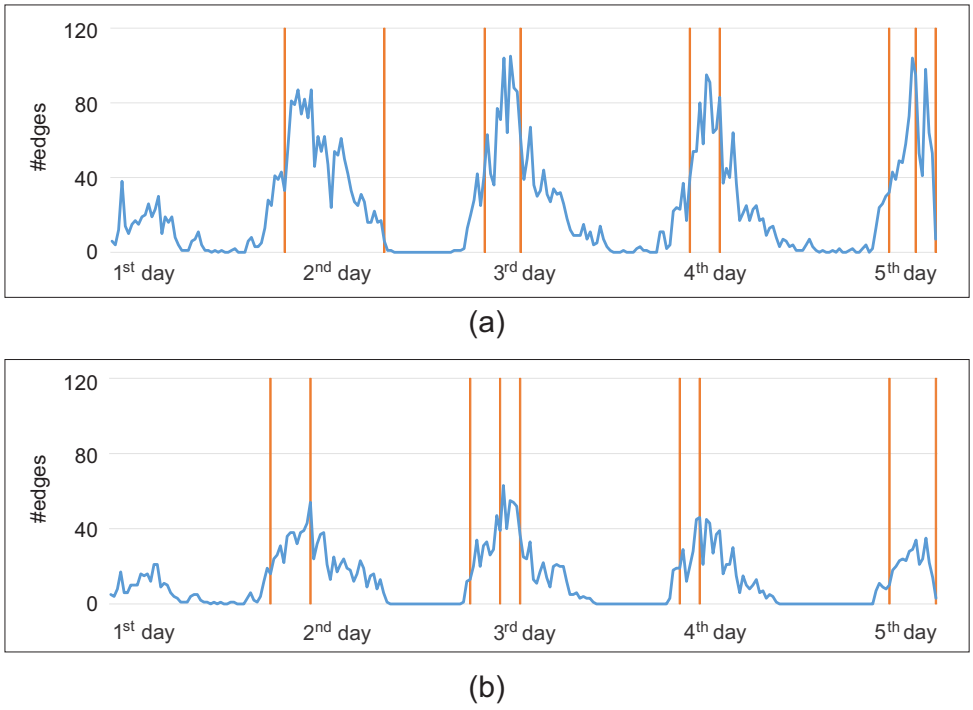

Figure 58 – Number of edges per timestamp (blue) and timestamps where changes were detected by PHT-FM for the Hospital network (orange). (a) Original network. (b) Network after SEVis sampling. SEVis discarded 48.2% of the edges while maintaining relevant characteristics of the original network measured by the KS distance (KS-d = 0.14) and PHT-FM (detection of changes in all days that presented changes in the original network). The SEVis configuration and the temporal resolution are the same as from Figure 56.

SEVis is suitable for streaming networks. On the other hand, MSV draws all nodes and edges of the entire network at once (ELZEN et al., 2013; LINHARES et al., 2019b). To illustrate SEVis application in layouts that are also suitable for streaming scenarios, Figure 59(a) shows temporal snapshots of the original Hospital network and Figure 59(b,c) the same snapshots after EOD and SEVis sampling, respectively. The nodes along the

snapshots of each method have fixed position over time (node-link diagram layout). Note that EOD is not suitable for streaming networks. Moreover, it is a MSV-based method, not suitable for other layouts without adaptation either. We used the edges accepted by EOD in its sampling process over the MSV layout as input for the node-link diagram construction. On the other hand, SEVis does not depend on specific layouts' characteristics (e.g., timeline vs animation-based layouts or length/positioning of edges) and thus is flexible to be used in any type of layout. Therefore, we applied our proposed SEVis directly in the animated node-link diagram. SEVis maintains more edges than EOD (Figure 59(b,c)). In $t = 21$, for instance, all four edges were discarded by EOD while SEVis discarded only one edge, keeping all edges incident to the highest degree node of this time. In $t = 47$, all nurses active in this time and one of the two patients were maintained by SEVis while EOD maintained only two nurses. In streaming network visualization using this layout and SEVis, the amount of stored information never becomes too large as long as the information of the current temporal window fits in primary memory. This happens because older nodes and edges are automatically discarded as the animation runs.

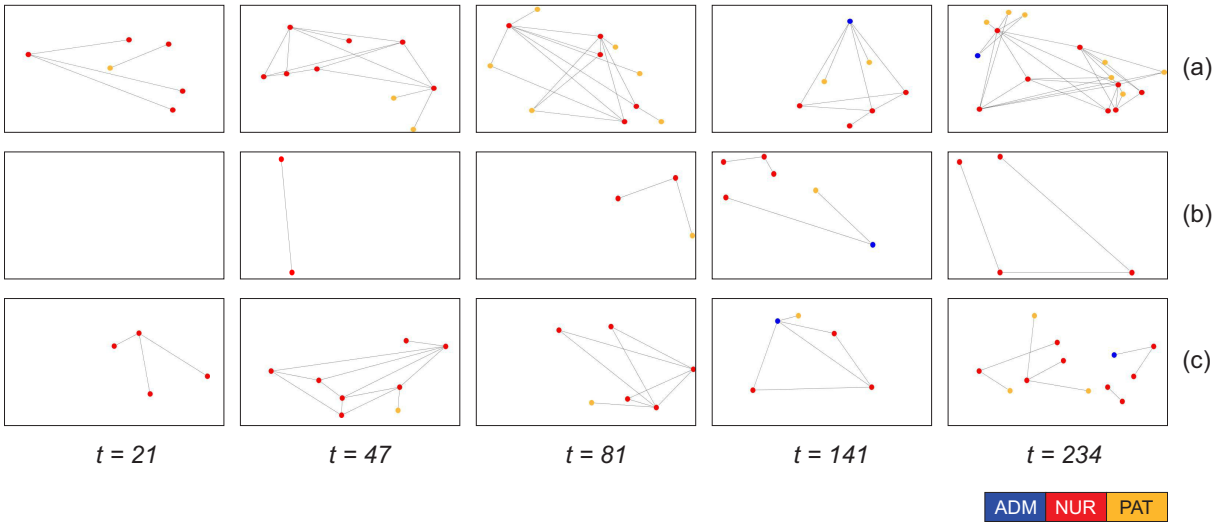

Figure 59 – Node-link diagrams for the Hospital network visualized in five different snapshots. (a) Network before sampling, (b) EOD, and (c) SEVis. The parameters are the same from Table 8, but with $w_{size} = 50$ for SEVis. For each sampling method, the nodes along each snapshot have fixed position over time. Each time refers to a 23-minute interval. EOD is not suitable for streaming networks.

6.2.4.3 Twitter$_s$ Network

This analysis considers a subset of the *Twitter* network (Section 2.1.3), hereafter named Twitter$_s$. It contains all nodes and edges from the first 156 (of 224) timestamps, and so it has 50,112 edges and 32,124 nodes. There are 321 edges per time on average,

in contrast to the *Hospital* network that contains only six edges per time on average. We have applied SEVis on such a large and dense network to evaluate SEVis in this scenario.

Figure 60 presents the MSV layout generated by CNO(Louvain, Degree, Degree) for this network. Recall that the topic related to the content of the tweet that each edge represents is known in advance (Sports, Celebrity, Corruption, Politics, Education, Security, or International), so different edge colors can be used to indicate different topics in the layout. Recall also that the color intensity is related to the number of overlapping edges in a way that, the more overlaps, the darker the colors. Due to the network size and density, the use of CNO node ordering alone does not reduce the number of overlaps to a level that allows an effective visual analysis (Figure 60(a)).

As illustrated in Figure 60(b), when applying SEVis ($w_{size} = 55$ and $k = 100$, empirically determined), the number of edges is reduced to 32,451, consequently reducing the number of overlaps (compare the color intensities in Figures 60(a-b)). However, the number of overlaps and the edge lengths remain high, impairing the visual analysis. This impairment does not imply in a SEVis low performance because a particular edge may be considered as an intra-community edge by SEVis (while considering only the most recent window of edges) and as an inter-community edge by CNO (while considering the aggregated network, i.e., all timestamps at once). As a consequence, it is possible to find long (according to CNO) and relevant (according to SEVis) edges on large networks. Running CNO per window along with SEVis would attenuate this issue and facilitate network community evolution analysis (e.g., merge/split of communities). MSV, however, does not meet such varying node positioning requirement.

As established by the visual information seeking mantra *"overview first, zoom and filter, then details-on-demand"* (SHNEIDERMAN, 1996), one should discard some information to be capable of analyzing particular regions of interests. In our context, we consider this principle by filtering out CNO inter-community edges – which are long and less relevant edges when considering the aggregated network – before applying more elaborate edge sampling approaches. By doing so, groups of nodes (CNO communities) emerge and now it is possible to interact with the layout (e.g., by *zooming in*) and analyze the topics been retweeted over time by particular communities (Figures 60(c,d)). There are communities still very polluted, and so edge sampling strategies, such as EOD and SEVis, can now be employed to improve readability in the layout produced after CNO inter-community edge filtering (Figure 60(c)). Contrary to EOD (Figure 60(e)), SEVis (Figure 60(f)) is capable of highlighting the most prevalent topics in all three communities indicated by Figure 60(d), with the advantage of a lower time processing. Such topics are the ones being retweeted over time by the $k = 100$ most relevant individuals in the network.

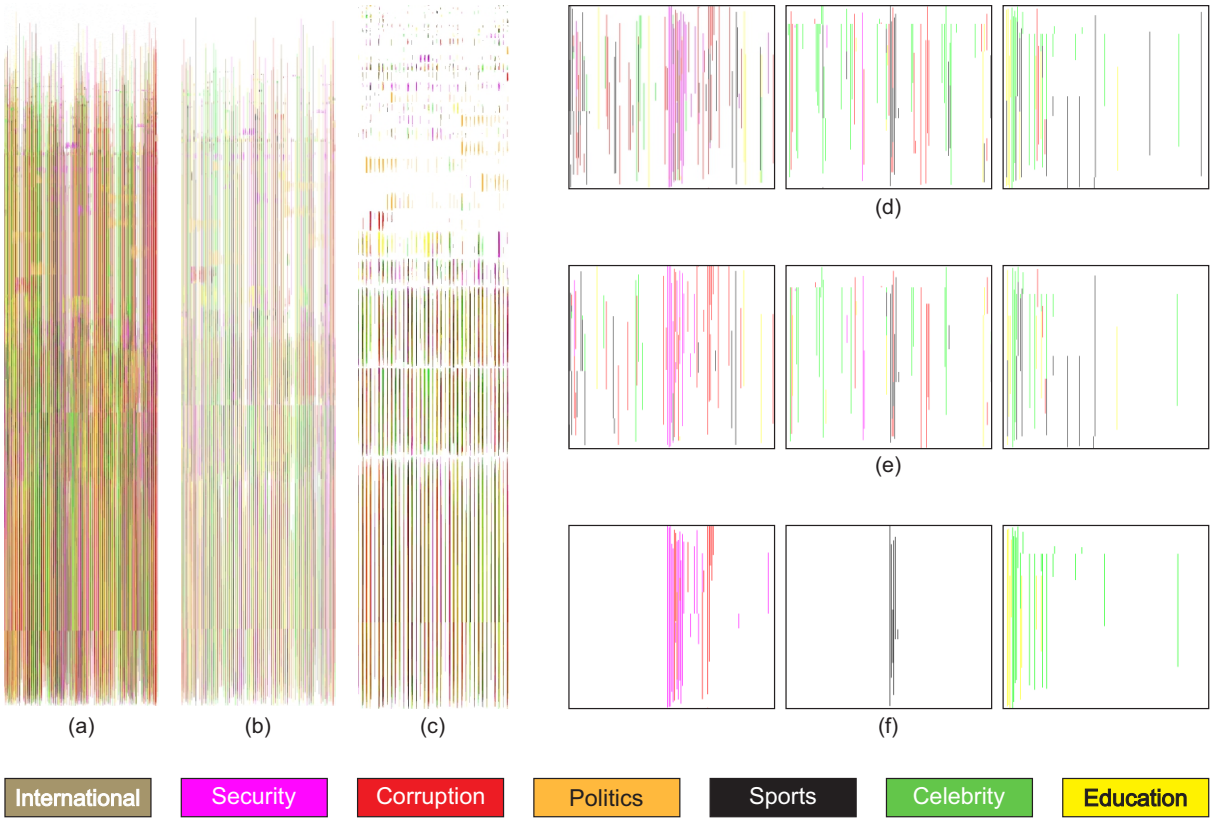

Figure 60 – MSV layout showing CNO communities after EOD and SEVis edge sampling for the $Twitter_s$ network. (a) Original network. (b) Network after SEVis ($w_{size} = 55$ and $k = 100$). (c) Original network after filtering out CNO inter-community edges. (d) Three CNO communities that can be found in (c). (e) The same three communities after applying EOD in (c). (f) The same three communities after applying SEVis in (c). Node ordering defined by CNO(Louvain, Degree, Degree). Each edge color indicates the topic related to the content of the tweet that such edge represents. Each time refers to a 1-hour interval (original temporal resolution).

6.3 Final Considerations

This chapter presented SEVis, a streaming edge sampling method for network analysis that selects edges while preserving the characteristics of the original network in terms of edge frequency and distribution changes. SEVis is flexible and can be used to improve a variety of layouts, as for example, MSV and node-link diagram. It is also capable of enhancing the analysis of temporal networks with distinct characteristics such as small/large, sparse/dense, non-streaming/streaming network data. We applied SEVis on random modular networks and real-world networks to evaluate its performance through experiments and comparative quantitative and visual analyses. The results indicated super performance of SEVis sampling against existing methods in the literature.

Any sampling strategy involves removal of information. As a consequence, relevant

information may be lost during the process. In the context of temporal networks, previously perceived patterns, groups of relevant nodes and bursts of connections, for example, may be decomposed or disappear. SEVis reduces such impairment by performing an edge sampling that takes into account the node relevance and the most relevant edges (i.e., intra-community edges), but the lost of relevant information still can happen in some cases.

Although SEVis improves layouts from several visualization strategies, the visual information, and consequently the level of visual clutter, may be different depending on the node positioning and temporal resolution being used. Methods that manipulate each of these dimensions (edge, node, and time) can be considered in the layout construction to allow effective visual analyses.

The SEVis configuration, i.e., the choice of which method should be used to detect communities, to verify whether the communities still represent the current state of the network, and to compute the node relevance, directly affects SEVis performance. The choice should take into account response time and quality. Methods with low response time may impair the SEVis application in streaming scenarios. Overall, SEVis is flexible and accommodates a variety of methods.

Finally, SEVis relies on network community detection and so the method has limitations regarding the absence of community structure, as for example, in the case of purely random networks. In this worst-case scenario, SEVis is expected to perform no worse than random sampling. A similar situation refers to seasonal occurrence of activity (e.g., a school network in which there is activity only in the morning). In such situation, some temporal windows may not have enough information to detect communities, thus affecting the quality of the sampling.

Chapter 7

Combining Dimensions

An efficient edge sampling method maintains the most relevant edges in the layout. However, such edges may be short or long depending on the adopted node positioning. In the same way, edge lengths are reduced by efficient node ordering strategies, but the amount of visual information may continue too large depending on the temporal resolution and number of edges. The combination of such methods affects the level of visual clutter, pattern identification, and decision making.

Along the experiments presented in this thesis, we usually combined two methods to evaluate our proposals. The layouts used to analyze the performance of the adaptive temporal resolution method (Chapter 4) adopted *Recurrent Neighbors* (LINHARES et al., 2017b) node ordering. When analyzing SEVis edge sampling (Chapter 6), the node positioning in the visual analysis was defined by CNO. In this chapter, we focus on combining different methods (one for each dimension) and analyzing the produced layouts. More specifically, we aim to answer questions such as *"Is there an improvement when using layouts produced by the combination of two methods (high-performance methods for two network dimensions and a naive approach for the other)?"* and *"Is there an improvement when using layouts produced by the combination of the three methods (high-performance methods for the three network dimensions)?"*.

This chapter is organized as follows. Section 7.1 describes the Susceptible-Infected (SI) infection dynamics model, used in the visual analyses. Section 7.2 presents case studies considering two real-world networks with distinct characteristics. Final considerations are discussed in Section 7.3.

7.1 Susceptible-Infected (SI) infection dynamics

This section is based on (LINHARES et al., 2019a), a theme-related publication that is the result of a collaboration occurred during this thesis development. In this chapter, we again rely on MSV to analyze the quality of the layouts produced by different combinations. However, besides the analysis of the network temporal structure, we now consider

infection spread dynamics taking place on the network as well. The temporal network characteristics, as, for example, the existence of periods of idleness as well as with bursts of interactions, make such networks useful to study epidemics (ROCHA; BLONDEL, 2013), and so the temporal visualization comprehends a useful tool in this context (MASUDA; LAMBIOTTE, 2016; LINHARES et al., 2019a). In fact, several real-world networks have been proposed to analyze infection spread in different environments, e.g., the *Primary School* and the *Hospital* networks, two of the four networks considered in the previous chapters.

For convenience, we consider a simple infection dynamics, named *susceptible-infected (SI)* model (BARRAT; BARTHÉLEMY; VESPIGNANI, 2008). In this model, a node can have one of two states at a particular timestamp t, susceptible (S) or infected (I). All nodes start susceptible, except for a node chosen to be initially infected (patient zero). At each subsequent timestamp, each interaction involving an infected node and one of its susceptible neighbors (adjacent nodes) has a probability p of infecting such neighbor. As an example, if $p = 1$, every susceptible node will certainly become infected after connecting to an infected node. In the SI model, an infected node does not recovery or turn susceptible again [1] (BARRAT; BARTHÉLEMY; VESPIGNANI, 2008; ROCHA; BLONDEL, 2013).

Figure 61 illustrates, considering $p = 1$ as the infection probability, how MSV and TAM can be employed for analyzing the SI dynamics upon a given temporal network (Figure 61(a)). In this case, MSV shows only the edges where the infection propagates, i.e., the transmission paths from infected to susceptible nodes (Figure 61(b)). This approach, along with a transparency effect on non-infected nodes, facilitates the identification of who infected whom and when the transmission occurred. Such information is useful to recognize the importance of particular nodes and act to regulate the infection spread. On the other hand, the TAM layout is useful to highlight the node state, and so the identification of how groups of nodes with the same state evolve is facilitated (Figure 61(c)). Both layouts could be merged into one, but the produced layout would probably impair the analysis of large networks due to the amount of visual information.

7.2 Case Studies

In this chapter, we are interested in analyzing the transmission path using MSV (Figure 61(b)). We have analyzed layouts generated by different methods' combinations as they highly affect the level of visual clutter and pattern identification. Two real-world temporal networks were considered in the study: the *Museum* network (a relatively small

[1] There are also models in which an (I)nfected node turn (S)usceptible again (SIS model) or (R)ecovers (SIR model) after being infected for t_r time steps (BARRAT; BARTHÉLEMY; VESPIGNANI, 2008).

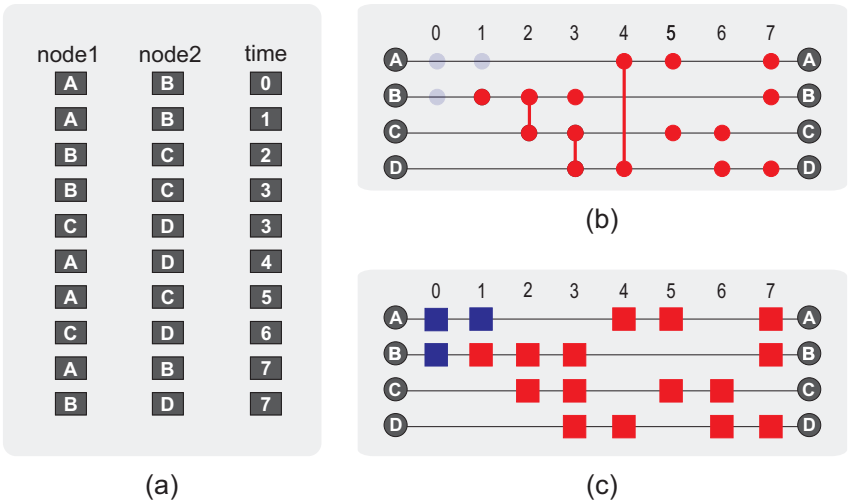

Figure 61 – Application of MSV and TAM layouts to analyze *Susceptible-Infected* infection dynamics. (a) List of edges. (b) MSV layout highlighting the transmission path. (c) TAM layout highlighting the node state. Infection probability $p = 1$. Blue color: Susceptible state. Red color: Infected state. Patient zero: node B at timestamp 1.

data set); and the *Sexual* network (a large data set). Both of them are described in Section 2.1.3.

For a given temporal resolution scale, the expression "Combination(x,y)" – or simply the tuple "(x,y)" – will be used to refer to the application sequence defined by, first, the employment of the node ordering method x, and second, the employment of the edge sampling method y. Note that the application of y before x would generate a completely different layout because of changes in the x's input edge set caused by y. In all cases, the methods x and y consider all edges, not only those highlighted in the layouts by the transmission tree.

We categorize each method used in our analyses as follows. Considering the temporal dimension, the original temporal resolution (Res. 1) is a naive and low quality choice to visually represent the network; our adaptive resolution, on the other hand, is a high performance method (see Chapter 4). For the node dimension, the node positioning defined by either *Appearance* (according to the timings of first connection) or *Degree* (according to the ascendant order of accumulated degree) produces a high level of visual clutter due to edge overlap and so has low performance; CNO, on the other hand, has been proved to be a better choice (see Chapter 5). Not least, SEVis is a high performance edge sampling method that improves the layout when compared with random sampling or no sampling at all (Chapter 6). The methods are configured as follows (parameter values empirically determined).

- Adaptive Resolution: fading factor equals to 0.99 ($FF = 0.99$) and window size of 50 ($w_{size} = 50$) or 100 timestamps ($w_{size} = 100$), depending on the analysis. See

Chapter 4 for details.

- CNO: *Louvain* (BLONDEL et al., 2008) as network community detection method and *Recurrent Neighbors* (LINHARES et al., 2017b) as community and node ordering strategy. See Chapter 5 for details.

- SEVis: by default, window size of 100 timestamps ($w_{size} = 100$) and $k = 0.25 \times |V|$ most relevant (i.e., most frequent) nodes considered by *Space-Saving* (METWALLY; AGRAWAL; ABBADI, 2005), where $|V|$ is the number of nodes in the network. Any change in these parameter values will be explicitly mentioned. The ratio to decide whether a new community detection must be executed, also using *Louvain*, is $t_r = 0.8$. See Chapter 6 for details.

- Random edge sampling: accepts an edge with probability $p_a = 0.5$.

7.2.1 Museum network

This analysis considers the *Museum* network, described in Section 2.1.3, and evaluates combinations involving: Res. 1 and Adaptive Resolution with $w_{size} = 50$ and $w_{size} = 100$ (temporal dimension); Appearance, Degree, and CNO node ordering (node dimension); None (without), Random, and SEVis edge sampling (edge dimension).

Figure 62 shows the cluttered-related measurement *number of intersections* (see Section 6.2.2) for each evaluated combination. The layouts adopting Res. 1 (Figure 62(a)) have a smaller number of intersections when compared with those from the adaptive resolution using $w_{size} = 50$ (Figure 62(b), avg(resolutions) = 2.44 $\pm$ 2.29) and $w_{size} = 100$ (Figure 62(c), avg(resolutions) = 2.86 $\pm$ 1.61). Edges from consecutive timestamps are grouped during the adaptive resolution computation, and so the overall number of timestamps is reduced. As a consequence, both edge overlap and the number of intersections increase. Although the layouts from Res. 1 have fewer intersections, they have more timestamps, which leads to large layouts (horizontally), thus impairing global pattern identification and requiring more screen space and, potentially, navigation tools such as *scrolling*. Navigation issues may then arise and affect the perception of temporal changes in the network, breaking the user's mental map.

When considering the same resolution scale, the combination *Appearance* for node ordering and *None* (without) for edge sampling – combination (Appearance, None) –, generates the layouts with the higher levels of visual clutter (higher number of intersections), followed by (Degree, None) and (CNO, None) – which reaffirms CNO high quality. When applying edge sampling strategies, the number of intersections is greatly reduced, as expected (see orange and gray bars in Figure 62). For Res. 1 (Figure 62(a)) and the adaptive resolution with $w_{size} = 50$ (Figure 62(b)), regardless of the node ordering, SEVis always outperforms random sampling, producing layouts that contain fewer inter-

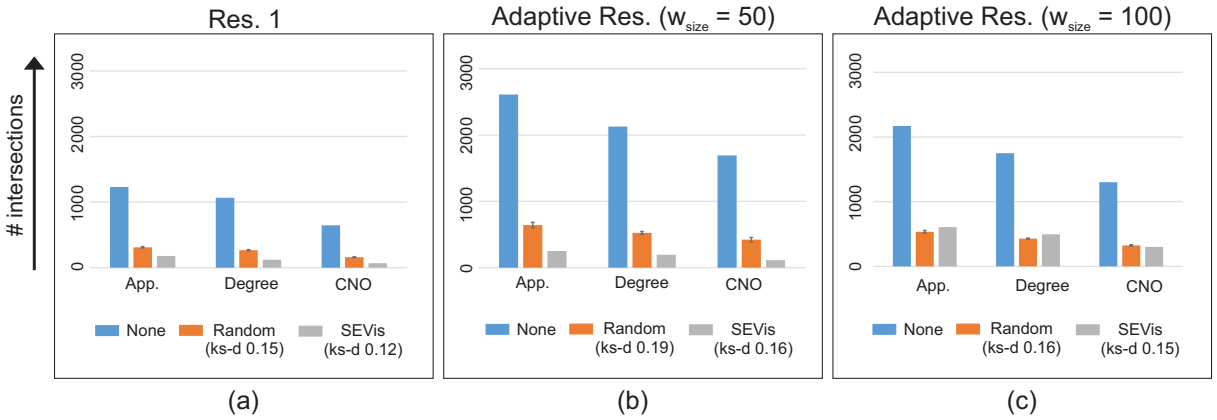

Figure 62 – Number of intersections per combination for the *Museum* network. (a) Res. 1. (b) Adaptive Resolution with $w_{size} = 50$. (c) Adaptive Resolution with $w_{size} = 100$. *App.* refers to the *Appearance* node ordering. Each color refers to an edge sampling approach (*None* means without sampling). Random sampling executed 10x for each node ordering. ks-d stands for *Kolmogorov-Smirnov* distance between the edge distribution before and after sampling. For both number of intersections and ks-d, lower values are better.

sections and better represent the original network (smaller *Kolmogorov-Smirnov* distance (ks-d) – see Section 6.2.2). When observing the layouts from the adaptive resolution with $w_{size} = 100$ (Figure 62(c)), however, SEVis produces more intersections than the random sampling for two node ordering methods (*Appearance* and *Degree*), i.e., SEVis maintained edges that are long when employing these ordering methods, thus increasing the number of intersections. When combined with CNO, however, SEVis outperforms the other combinations regardless of the temporal resolution being used: CNO reduces the number of intersections by repositioning nodes, which implies in less visual clutter and easier pattern identification, and SEVis improves the layout even more by discarding less relevant edges while preserving the characteristics of the network before sampling.

Figure 63 presents four layouts generated by different combinations using the adaptive resolution with $w_{size} = 100$ (Figure 62(c)). This configuration was chosen because the application of SEVis along with the adaptive resolution with $w_{size} = 100$ resulted in more intersections than SEVis along with the adaptive resolution with $w_{size} = 50$ (see gray bars in Figures 62(b,c)). Our goal is to evaluate SEVis under this circumstance, i.e., with such a high intersection level. The layout from (Appearance, None), shown in Figure 63(a), is dense and has too many intersections, thus impairing pattern identification. When applying SEVis (combination (Appearance, SEVis), Figure 63(b)), several edges of the network are discarded (1,749 out of 4,600), but the maintained edges remain too long, and so the layout readability is almost the same (although a few groups of relevant nodes are now highlighted – see red brackets). By repositioning nodes with CNO in the network without sampling (combination (CNO, None), Figure 63(c)), the analysis

is greatly improved: groups of nodes are highlighted (red brackets) and a time interval that with (Appearance, None) appeared to have a high level of activity is, in fact, an interval with low activity (see the red bar in Figure 63(c) and the respective time interval in Figure 63(a)). Applying SEVis after CNO (combination (CNO, SEVis), Figure 63(d)) produces a cleaner layout, with easier and faster identification of relevant groups of nodes (see red brackets).

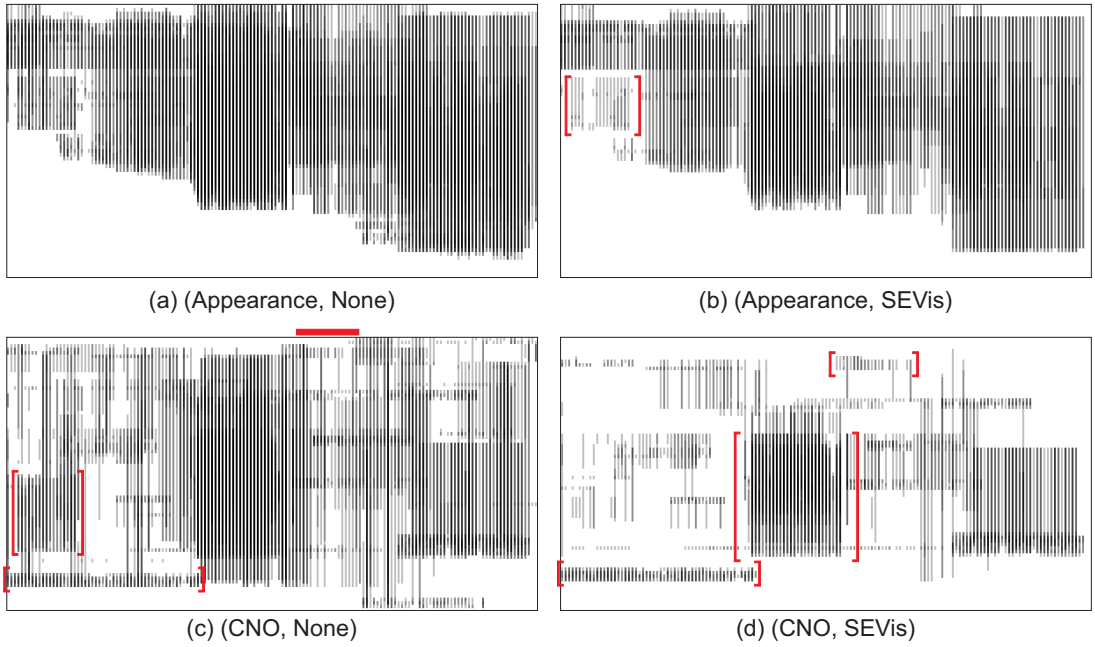

Figure 63 – MSV layouts generated by different combinations for a portion of the *Museum* network using the adaptive resolution with $w_{size} = 100$. (a) Combination (Appearance, None). (b) (Appearance, SEVis). (c) (CNO, None). (d) (CNO, SEVis). Time interval: from timestamp $t = 100$ to $t = 250$ (which corresponds to the interval from $t = 100$ to $t = 769$ in Res. 1).

The impact of different combinations in an epidemics visual analysis is shown in Figure 64. For this evaluation, we first simulated an infection spread applying the SI model in the original network (Res. 1 and without edge sampling) with infection probability $p = 0.1$. The patient zero is the node with the highest degree in the aggregated network (i.e., when considering all timestamps at once) in the first timestamp it appears (node id 25/timestamp $t = 52$). By comparing combination (Appearance, None) from Res. 1 (Figure 64(a)) with the combination (Appearance, None) from the adaptive resolution with $w_{size} = 100$ (Figure 64(b)), one can notice that the adaptive resolution maintains the characteristics of the edge distribution from Res. 1 while reducing the number of timestamps (1,312 vs 569 for representing the entire network). This leads, for instance, to faster identification of the epidemic outbreak and other high activity periods (black bars in Figure 64(b)).

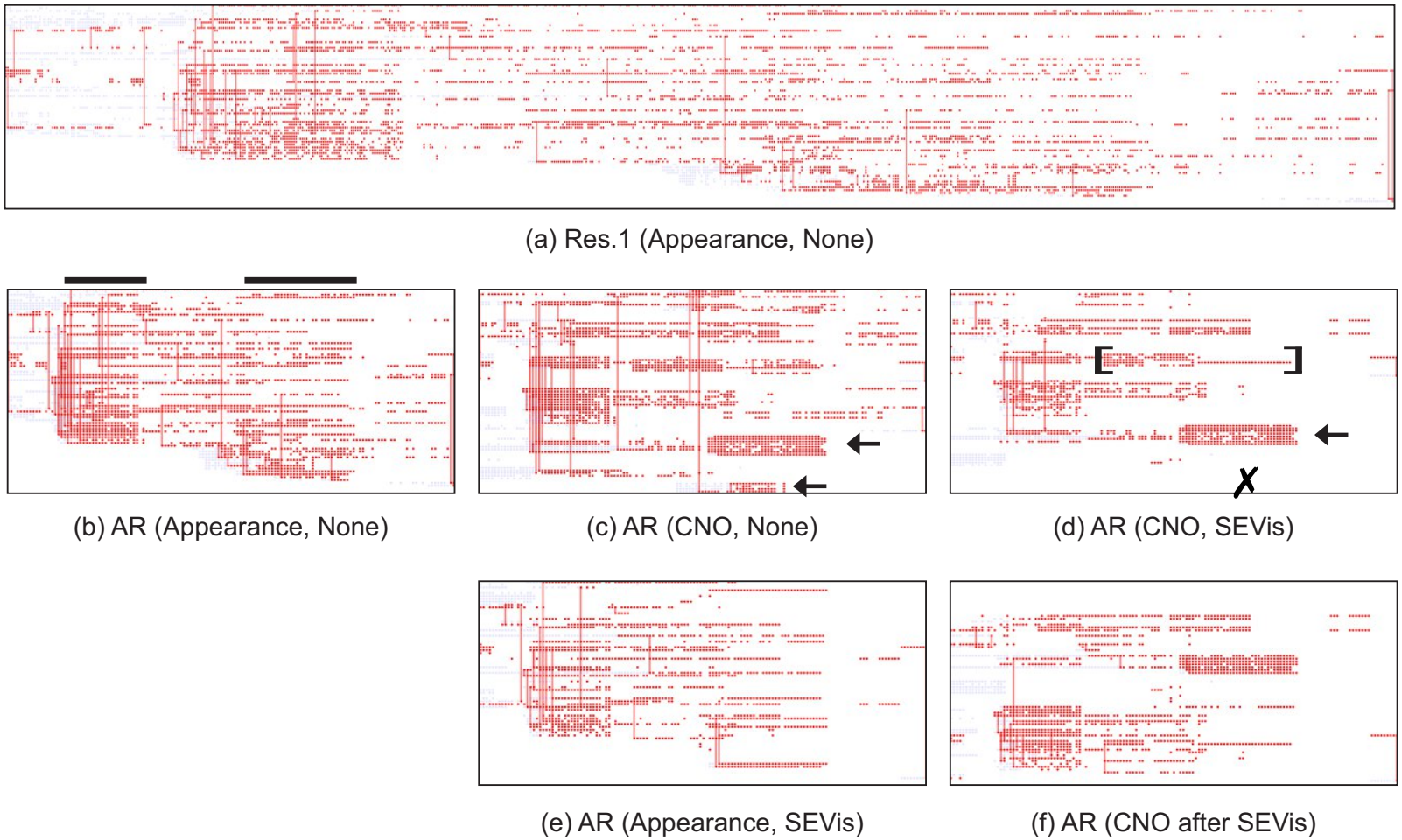

Figure 64 – Impact of different combinations in an epidemics visual analysis using the *Museum* network. (a) Res. 1, combination (Appearance, None). (b-f) Adaptive resolution with $w_{size} = 100$ (AR). (b) (Appearance, None). (c) (CNO, None). (d) (CNO, SEVis). (e) (Appearance, SEVis). (f) CNO applied in the layout from (e), i.e., CNO after SEVis. First application sequence: (a) → (b) → (c) → (d). Second application sequence: (a) → (b) → (e) → (f). Time interval: from timestamp $t = 220$ to $t = 870$ (in Res. 1).

By adopting the adaptive resolution with $w_{size} = 100$ and moving from (Appearance, None) to (CNO, None), as illustrated in Figures 64(b,c), the identification of groups of nodes is now possible (see arrows in Figure 64(c)). Such groups, that are obtained by CNO, correspond to network communities, i.e., groups of nodes that connect more often among themselves than to other nodes (LANCICHINETTI; FORTUNATO, 2009). Network communities in the context of an infection dynamics may indicate family groups, for instance, and so one may be interested in analyzing intra-community infection spread (BONACCORSI et al., 2014). When applying SEVis (combination (CNO, SEVis), Figure 64(d)), only the most relevant nodes and edges are maintained, and so a CNO community may continue the same (see arrow in Figure 64(d)), completely disappear (see 'X' symbol), or change (see brackets). If we move from (Appearance, None) to (Appearance, SEVis) instead of to (CNO, None), as shown in Figures 64(b,e), the layout becomes cleaner but the identification of groups is infeasible. Only when running CNO, such identification is possible (Figure 64(f)). Note that the node positioning is different when comparing Figures 64(d,f). After sampling, the set of edges considered by the CNO community detection step is a subset of the original one (before sampling). Different network communities are detected and therefore different node ordering are produced.

Overall, the combination of the adaptive resolution along with CNO (without SEVis) improved pattern identification in this network (Figure 64(c)). Applying SEVis without CNO, on the other hand, did not present the same effectiveness, even considering the adaptive resolution (Figure 64(e)). In summary, there are pairwise combinations that enhance the visual analysis, but not all of them. Not least, the combination of all three methods improved layout readability regardless of the order in which the node ordering and the edge sampling methods were employed (Figures 64(d,f)).

7.2.2 Sexual network

This analysis considers the *Sexual* network, described in Section 2.1.3. Figure 65 shows different combinations over the *Sexual* network. For this evaluation, we applied the SI model in the original network (Res. 1 and without edge sampling) with infection probability $p = 0.25$. The patient zero is an arbitrary node (node id 87) in the first timestamp it appears ($t = 3$). Given the network size, we did not adopt the node with the highest degree as patient zero. For the same reason, all analyses with SEVis adopt $k = 0.005|V|$, i.e., SEVis considers all edges incident to the top-60 more relevant (frequent) nodes. In the layouts, the intensity of the edge color is related to the number of overlapping edges and intersections (the darker the color, the higher the number of overlaps and intersections). By comparing (Appearance, None) with (CNO, None) in Res. 1 (Figures 65(a,b)), one may see that CNO greatly reduces the number of edge overlaps (lighter colors). The identification of patterns in the layout, however, remains difficult. Even when applying CNO and SEVis, global pattern identification is not optimized (Figure 65(c)).

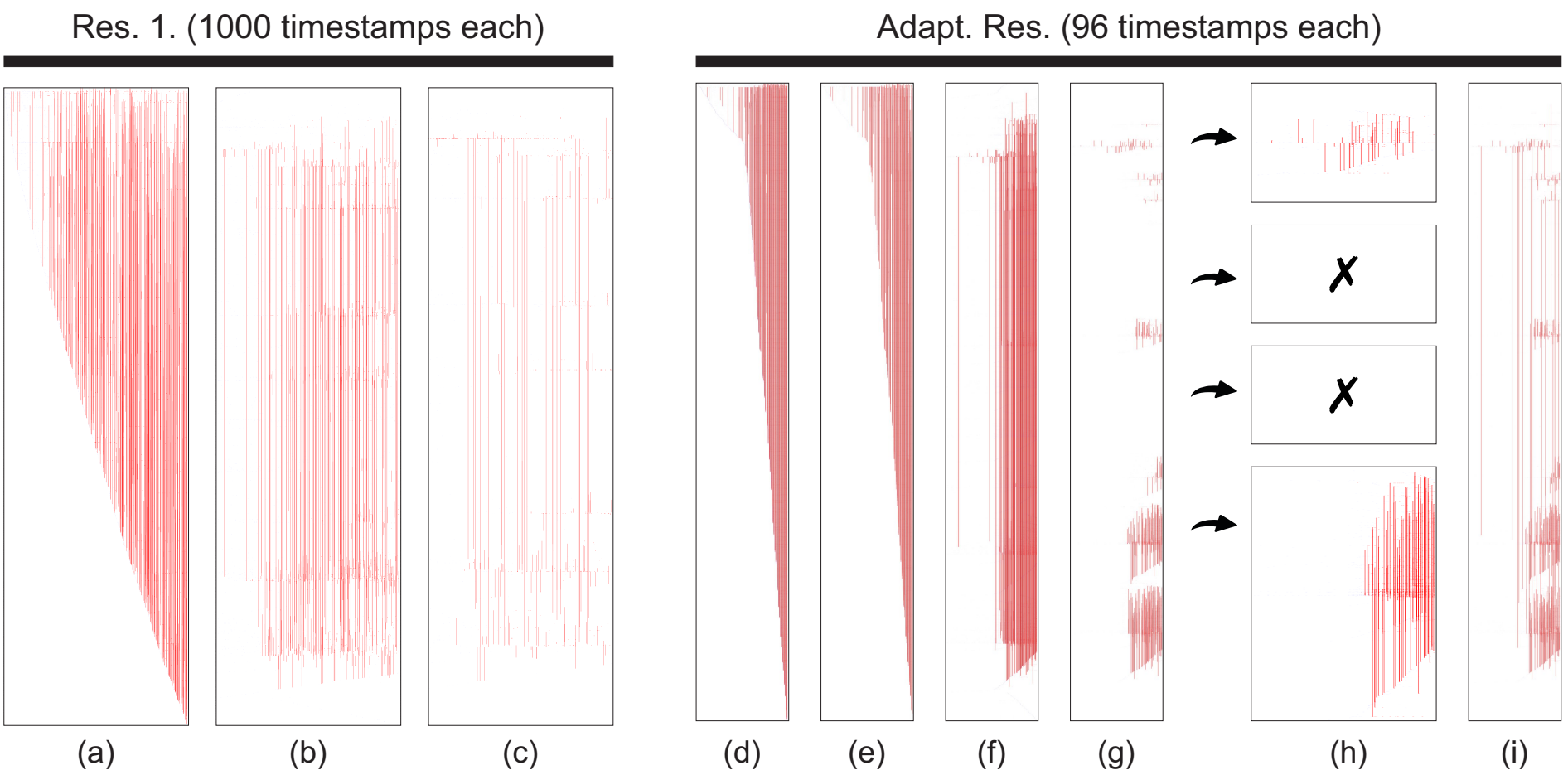

Figure 65 – Impact of different combinations in an epidemics visual analysis using the *Sexual* network. (a-c) Res. 1 (1,000 timestamps). (a) (Appearance, None). (b) (CNO, None). (c) (CNO, SEVis $w_{size} = 100$). (d-g) Adaptive resolution with $w_{size} = 50$ (96 timestamps). (d) (Appearance, None). (e) (Appearance, SEVis $w_{size} = 100$). (f) (CNO, None). (g) (CNO, SEVis $w_{size} = 100$). (h) *Zooming in* particular groups from (g). (i) (CNO, SEVis $w_{size} = 40$).

Because of the network density, the adaptive resolution method with $w_{size} = 50$ and $FF = 0.99$ redistributed all edges in 96 timestamps (avg(resolutions) $= 24.52 \pm 5.55$). Considering that the first w_{size} timestamps, used in the adaptive resolution computation, correspond to the method's cold start (see Section 6.1 for details), 950 timestamps of Res. 1 were converted in only 46 timestamps in the adaptive resolution. The layout produced by the combination (Appearance, None) considering this resolution is shown in Figure 65(d). Note that approximately the first half of the layout (left-middle) contains too few edges (cold start) whilst the second half (middle-right) contains all the other edges in a high edge overlap level (dark colors). So many overlaps involving long edges lead to a number of intersections so elevated that the visual analysis becomes unfeasible without complementary strategies that promote further edge reduction or removal (e.g., CNO along with SEVis).

For the adopted adaptive resolution scale, neither the combination (Appearance, SEVis) nor (CNO, None) improves the overall layout readability (Figures 65(e,f)). Only when combining all three methods (adaptive resolution plus (CNO, SEVis)), particular groups of nodes and edges are revealed in the *Sexual* network. Given the new number of timestamps (96), running SEVis with $w_{size} = 100$ means that the entire network is considered in the execution. In this case, there is a match between the communities detected by both CNO and SEVis (considering both using *Louvain*), so all CNO inter-community edges are discarded by SEVis. Recall that CNO also allows filtering out inter-community edges (see Section 5.2.1.3 for details). The differences between the layouts generated by such CNO edge filtering and (CNO, SEVis), in this case, rely on SEVis 2^{nd} filter, responsible for discarding edges involving non-relevant nodes. In a more detailed perspective, the analysis of such discrepancies is analogous to the one from Figures 64(c,d).

The aforementioned situation is a particular case in which (i) the same network community detection method is employed in both CNO and SEVis, and (ii) the advantages of SEVis for streaming scenarios are not considered, i.e., the entire network is considered by both methods. After all, the groups of nodes and edges that were maintained in the layout (Figure 65(g)) may be further analyzed using interactive tools, such as *zoom* and *pan* (Figure 65(h)), that allow to explore, e.g., nodes inside a particular group or perform cross-comparison between groups. Groups of non-infected nodes and groups considered as non-relevant by SEVis are discarded and therefore not visible (see 'X' symbols in Figure 65(h)).

When applying SEVis using smaller windows, there is no match between the communities from CNO and SEVis, and so edges maintained by SEVis may connect nodes positioned far from each other, thus increasing the number of intersections and polluting the layout. Figure 65(i) shows the layout produced by CNO and SEVis using $w_{size} = 40$. Although small groups may be lost due to edge overlap (in comparison with Figure 65(g)), it is now possible to identify relevant connections between the visible groups, which fa-

cilitates contact tracing and cross-comparisons. The mentioned interactive tools would benefit this analysis as well.

7.3 Final Considerations

This chapter presented two case studies in which, by simulating infection spread dynamics in real-world networks, we were capable of showing how the combination of different methods affects the layout and, consequently, pattern identification and decision making. Examples of patterns include tracking intra-community infection spread, source identification, and identifying relevant infected nodes (information that could be used, e.g., to regulate the spread through vaccination schemes).

We have considered real-world networks with distinct characteristics in the presented case studies. The *Museum* network is a relatively small data set (72 nodes and 5.32 edges per timestamp on average) that contains more timestamps than nodes. The *Sexual* network is a large data set (12,157 nodes and 34.06 edges per timestamp on average) that contains more nodes than timestamps. We cannot generalize the findings presented in this chapter as a single parameter modification in any method would generate a completely different layout. Moreover, other tasks and networks, combinations involving other methods, and other application sequences (e.g., executing SEVis first and then running the adaptive resolution) were not tested as well. Nevertheless, we are now capable of responding to the questions presented at the beginning of this chapter for the performed evaluation:

"Is there an improvement when using layouts produced by the combination of two methods (high-performance methods for two network dimensions and a naive approach for the other)?"
Answer: It depends on the network characteristics and the combination/parametrization being considered. For the *Museum* network, the combination of the adaptive resolution and CNO (without SEVis) greatly improved layout readability (Figure 64(c)). The combination of the adaptive resolution and SEVis (without CNO and therefore using *Appearance*), on the other hand, did not present the same effectiveness (Figure 64(e)). In essence, an edge sampling method should be considered as a complementary strategy, always associated with the employment of a high performance node ordering.

"Is there an improvement when using layouts produced by the combination of the three methods (high-performance methods for the three network dimensions)?"
Answer: Yes. To a greater or lesser extent, this happened for both networks. When analyzing the *Sexual* network, only the combination of all three methods (adaptive resolution, CNO, and SEVis) with the adopted parametrization highlighted existing patterns (Fig-

ures 65(g-i)). The evaluated pairwise combinations were not effective (Figures 65(c,e,f)).

Lastly, contrary to CNO, that requires the aggregated network, SEVis and the temporal resolution method are suitable for streaming networks as they adopt only the more recent data through windows over time. The "gap" that exists between the static CNO ordering and SEVis dynamic execution may result in discrepancies in the community detection, leading to long edges in the MSV layout (recall Figure 65(i)). Running both SEVis and CNO in such a dynamic manner would naturally attenuate this situation as CNO would change the node positioning at each window, thus approximating nodes and facilitating network community evolution analysis (e.g., birth/death, merge/split of communities). The drawback is that MSV does not allow changes in the node positioning over time because of the visual complexity and user's mental map preservation. The proposal of a new layout to address this issue is left as future work.

Chapter 8

Conclusion

Visualization strategies are useful for temporal network analysis since they provide means for quick assessment and support for decision making. However, depending on the visualization strategy being used and the network size, the visual clutter may severely impair the analysis, hiding meaningful temporal patterns and resulting in misleading conclusions. Furthermore, real-time network data (e.g. online media, financial data, high-throughput biological and neuronal activity data) have been generated in increasing volume and speed. Such challenges call for novel and efficient methods to process streaming network data.

For an effective visual analysis, different network dimensions (*node*, *edge*, and *time*) can be manipulated. An efficient edge sampling approach, for example, can maintain the most relevant edges in the layout. However, such edges may be short or long depending on the adopted node positioning. This combination affects the level of visual clutter and, consequently, pattern identification and decision making. Considering this context, we proposed methods to manipulate each of the network dimensions. Our focus in on streaming scenarios, but, as demonstrated in the presented case studies, our proposals are also applicable in non-streaming scenarios and can be used when dealing with temporal networks with distinct characteristics such as small/large and sparse/dense network data.

Considering the time dimension, Chapter 4 details a method to adapt the network temporal resolution scale in an automatic fashion, according to the different levels of node activity in the network over time. We conducted experiments using two real-world networks and the results show the identification of patterns that would be difficult to perceive when using the original static resolution.

In the MSV layout, nodes have fixed positioning over time due to mental map preservation. In streaming scenarios, an incremental node ordering would change node positioning in time as they gain or lose relevance, and so this MSV constraint is a problem. Despite this, relevant regions of interest can be identified during streaming network analyses. Such regions can be treated as non-streaming sub-networks and can be further analyzed using any node ordering method. Chapter 5 presented *Community-based Node Ordering*

(CNO), a visual scalable node ordering method for network visualization that combines community detection with node ordering techniques to enhance the identification of visual patterns. We performed visual and quantitative analyses using two real-world networks with different characteristics to show that the produced layout facilitates the identification of patterns that would otherwise be difficult to see. As CNO uses community structure information, we have also developed a visualization method that allows the analysis and comparison of two community detection algorithms. We performed case studies involving two popular methods and two real-world networks. As a result, the behavior of both algorithms in each case study is shown, helping the user in choosing the more adequate method for the network under analysis.

Chapter 6 presented *Streaming Edge Sampling for Network Visualization* - SEVis, a streaming edge sampling method that discards less relevant edges while maintaining the characteristics of the original network in terms of edge frequency and distribution changes. It can be applied to a variety of layouts to enhance both streaming and temporal network analyses. We evaluated SEVis performance using synthetic and real-world networks through quantitative and visual analyses. The results indicate a higher performance of SEVis regarding clutter reduction and pattern identification when compared with other sampling methods.

Finally, we have shown along all chapters, but mainly in Chapter 7, the importance of considering different dimensions in the layout construction. For the evaluated scenarios, the layouts produced by the combination of high-performance methods (node ordering, edge sampling, and temporal resolution) provide better visual analyses when compared with the adoption of a single method.

8.1 Main Contributions

Our main contributions can be summarized as follows.

- A novel and adaptive temporal resolution method for streaming networks that considers local levels of node activity over time.
- A novel and visual scalable node ordering method (CNO), useful for the analysis of large networks.
- A novel edge sampling method (SEVis), suitable for streaming networks, that discards less relevant edges while maintaining the characteristics of the original network in terms of edge frequency and distribution changes.
- A method for evaluating the performance of different network community detection algorithms through visual analysis.

- Presentation of case studies in which the combination of the manipulation proposals improved the visual analysis.

- An extension of the software DyNetVis (LINHARES et al., 2017b) that incorporates our methods and also competing ones (e.g., BVC temporal resolution (WANG et al., 2019) and EOD edge sampling (ZHAO et al., 2018)). This extension will be freely available at <www.dynetvis.com>.

8.2 Directions for Future Research

As presented in this thesis, MSV is a timeline visualization strategy suitable for temporal network visualization. The number of edges in a stream, however, is unknown and possibly unbounded (AHMED; NEVILLE; KOMPELLA, 2013), so there is no guarantee that the screen space and primary memory capacity are enough to draw all timestamps in the MSV layout.

Streaming data visualization is a challenging research field (MANSMANN et al., 2012; KRSTAJIĆ; KEIM, 2013) and, more specifically, streaming network visualization has recently been considered in literature (GRABOWICZ; AIELLO; MENCZER, 2014; CRNOVRSANIN; CHU; MA, 2015; SARMENTO; CORDEIRO; GAMA, 2015a). For non-streaming temporal networks, most techniques can be classified as using either animation (e.g., animated node-link diagram) or static timeline (e.g., MSV); only a few studies, such as (BURCH; WEISKOPF, 2014; SALLABERRY; MUELDER; MA, 2013; BACH et al., 2015), combine both of them. In fact, the combination of animation and timeline has not been fully explored (BECK et al., 2017).

By incorporating animation in the MSV layout to deal with the unknown number of timestamps, and the network manipulation methods that we proposed, we believe that such hybrid MSV-based layout may contribute for better visual analysis of streaming networks. The development of this layout must take into account several factors that affect the quality of the layout, in terms of pattern identification and mental map preservation. Examples include the animation transition speed and the number of timestamps that will be simultaneously shown. After development, an user study considering all these aspects would be valuable to assess the visualization quality.

The methods proposed in this thesis can also be further improved. Some ideas for future work are presented below.

Adaptive Temporal Resolution Method. One interesting future work is related to the measurement of how much information is lost when the network temporal resolution is changed. This could be done by analyzing information theory-based measures, such as entropy, and would be of great value to assess the method's quality. From a visualization perspective, one may perform user evaluation to validate the

method considering the quality of the produced layout. Besides, the choice of both window size and fading factor value directly affects the layout. These are currently user-dependent parameters that can be chosen with an initial exploratory analysis, and automate them is an interesting future work. Not least, the method performance can be also analyzed in other network visualization strategies.

CNO. Develop layouts that take into account community evolution (CAZABET; ROSSETTI, 2019). The application of other node reordering methods, such as *Optimized MSV* (ELZEN et al., 2014), and other community detection algorithms, may also improve CNO, so new experiments could be performed. At last, one can perform user evaluation to analyze CNO regarding mental map preservation and perceptual complexity.

Visual Analysis for Evaluating Community Detection Algorithms. Improve the study of community visual analysis, specially with experiments based on user tasks, to identify whether the best method according to statistical measures presents the best visual experience. In addition, one can extend the analysis to include other network community detection algorithms and also involve networks with more nodes and edges, or networks from other domains.

SEVis. Apply the method in layouts other than MSV and node-link diagram and analyze if (and how) SEVis benefits specific user tasks. As SEVis discards network elements (nodes and edges), a second future work refers to the measurement of how much information is lost after sampling (e.g., by evaluating the entropy). This evaluation represents a relevant aspect to ensure the method's performance. At last, SEVis can be used as a general method to summarize networks that may be analyzed by other means, not only by visual analysis. Thus, evaluating SEVis in cases not involving network visualization is also an interesting theme for future work. For further analysis using *Page-Hinkley* test as change detection method, one should consider a better adjustment considering the sensibility of the parameters δ and λ, e.g., by testing different combinations (SEBASTIÃO et al., 2013) or automating δ (SEBASTIÃO; FERNANDES, 2017).

8.3 Bibliographical Contributions

This thesis generated the following bibliographical contributions:

Journal publications:

1. Linhares, C.D.G., **Ponciano, J.R.**, Pereira, F.S., Rocha, L.E., Paiva, J.G.S., Travençolo, B.A.N.: *A scalable node ordering strategy based on community structure*

for enhanced temporal network visualization. Computers & Graphics 84, 185-198 (2019). Available at <https://doi.org/10.1016/j.cag.2019.08.006>.

2. Linhares, C.D.G., **Ponciano, J.R.**, Pereira, F.S., Rocha, L.E., Paiva, J.G.S., Travençolo, B.A.N.: *Visual analysis for evaluation of community detection algorithms.* Multimedia Tools and Applications 79(25), 17645-17667 (2020). Available at <https://doi.org/10.1007/s11042-020-08700-4>.

3. **Ponciano, J.R.**, Linhares, C.D.G., Melo, S. L., Lima, L.V., Travençolo, B.A.N.: *Visual analysis of contact patterns in school environments.* In: Informatics in Education 19(3), 455–472 (2020). Available at <https://doi.org/10.15388/infedu.2020.20>.

Under revision:

1. **Ponciano, J.R.**, Linhares, C.D.G., Faria, E.R., Travençolo, B.A.N.: *An Online and Nonuniform Timeslicing Method for Network Visualisation.* In: Computers & Graphics (journal).

2. **Ponciano, J.R.**, Linhares, C.D.G., Rocha, L.E., Faria, E.R., Travençolo, B.A.N.: *A Streaming Edge Sampling Method for Network Visualization.* In: Knowledge and Information Systems (journal).

Besides the aforementioned contributions, some collaborations that occurred during this thesis development resulted in the following theme-related publications:

1. Linhares, C.D.G., **Ponciano, J.R.**, Paiva, J.G.S., Travençolo, B.A.N., Rocha, L.E.: *Visualisation of structure and processes on temporal networks.* In: Temporal Network Theory. Cham: Springer International Publishing, 2019. p. 83–105. ISBN 978-3-030-23495-9.

2. Linhares, C.D.G., **Ponciano, J.R.**, Rocha, L.E., Paiva, J.G.S., Travençolo, B.A.N.: *Análise temporal de uma rede de contato hospitalar utilizando técnicas de visualização de informação.* In: [S.l.]: XVII Workshop de Informática Médica, 2017.

3. Pereira, F.S.F., Linhares, C.D.G., **Ponciano, J.R.**, Gama, J., de Amo, S., and Oliveira, G.M.B. (2019). *Uma Análise sobre a Evolução das Preferências Musicais dos Usuários Utilizando Redes de Similaridade Temporal.* iSys-Revista Brasileira de Sistemas de Informação, 12(3), 94-115.

4. Pereira, F.S.F., Linhares, C.D.G., **Ponciano, J.R.**, Gama, J., de Amo, S., and Oliveira, G.M.B. (2018). *That's my jam! uma análise temporal sobre a evolução das preferências dos usuários em uma rede social de músicas.* In 7o Brazilian Workshop on Social Network Analysis and Mining (BraSNAM 2018), volume 7, pages 160–171. SBC.

Other collaborations resulted in two submitted articles. In the first one, we describe in details the software DyNetVis in terms of motivation and significance, requirements, functionalities, and scientific impact. The second one presents a user study conducted to compare four layouts (*animated node-link diagram*, *animated matrix*, *MSV*, and *TAM*) according to tasks defined by a taxonomy for network evolution analysis (AHN; PLAISANT; SHNEIDERMAN, 2014). With this study, we aim to identify whether a particular layout is more suitable for a particular user task than the others.

1. Linhares, C.D.G., **Ponciano, J.R.**, Rocha, L.E., Paiva, J.G.S., Travençolo, B.A.N.: *DyNetVis - An interactive software to visualize structure and epidemics on temporal networks* In: ASONAM 2020.

2. Linhares, C.D.G., **Ponciano, J.R.**, Rocha, L.E., Paiva, J.G.S., Travençolo, B.A.N.: *User evaluation of temporal network visualization techniques* In: International Journal of Human–Computer Studies.

Bibliography

ACKERMANN, M. R.; MÄRTENS, M.; RAUPACH, C.; SWIERKOT, K.; LAMMERSEN, C.; SOHLER, C. StreamKM++: A clustering algorithm for data streams. **ACM Journal of Experimental Algorithmics**, Association for Computing Machinery, New York, NY, USA, v. 17, maio 2012. ISSN 1084-6654. Disponível em: <https://doi.org/10.1145/2133803.2184450>.

AGGARWAL, C. C. **Data Streams: Models and Algorithms (Advances in Database Systems)**. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387287590.

AGGARWAL, C. C.; YU, P. S.; HAN, J.; WANG, J. A framework for clustering evolving data streams. In: **VLDB '03: Proceedings of the 29th international conference on Very large data bases**. San Francisco: Morgan Kaufmann, 2003. p. 81 – 92. ISBN 978-0-12-722442-8. Disponível em: <https://doi.org/10.1016/B978-012722442-8/50016-1>.

AGGARWAL, C. C.; ZHAO, Y.; YU, P. S. Outlier detection in graph streams. In: **2011 IEEE 27th International Conference on Data Engineering**. Washington, DC, United States: IEEE Computer Society, 2011. p. 399–409. ISBN 978-1-4244-8958-9. Disponível em: <https://doi.org/10.1109/ICDE.2011.5767885>.

AHMED, N. K.; DUFFIELD, N.; WILLKE, T. L.; ROSSI, R. A. On sampling from massive graph streams. **Proceedings of the VLDB Endowment**, VLDB Endowment, v. 10, n. 11, p. 1430–1441, ago. 2017. ISSN 2150-8097. Disponível em: <https://doi.org/10.14778/3137628.3137651>.

AHMED, N. K.; NEVILLE, J.; KOMPELLA, R. Network sampling: From static to streaming graphs. **ACM Transactions on Knowledge Discovery from Data**, ACM, New York, NY, USA, v. 8, n. 2, p. 7:1–7:56, jun. 2013. ISSN 1556-4681. Disponível em: <https://doi.org/10.1145/2601438>.

AHN, J.-w.; PLAISANT, C.; SHNEIDERMAN, B. A task taxonomy for network evolution analysis. **IEEE Transactions on Visualization and Computer Graphics**, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 20, n. 3, p. 365–376, mar. 2014. ISSN 1077-2626. Disponível em: <http://dx.doi.org/10.1109/TVCG.2013.238>.

AHN, Y.-Y.; BAGROW, J. P.; LEHMANN, S. Link communities reveal multiscale complexity in networks. **Nature**, Nature Publishing Group, v. 466, n. 7307, p. 761–764, June 2010. Disponível em: <https://doi.org/10.1038/nature09182>.

AL-KHATEEB, T.; MASUD, M. M.; KHAN, L.; AGGARWAL, C.; HAN, J.; THURAISINGHAM, B. Stream classification with recurring and novel class detection using class-based ensemble. In: **ICDM '12: Proceedings of the 2012 IEEE 12th International Conference on Data Mining**. Washington, DC, United States: IEEE Computer Society, 2012. p. 31–40. Disponível em: <https://doi.org/10.1109/ICDM.2012.125>.

ALBERT, R.; BARABÁSI, A.-L. Statistical mechanics of complex networks. **Reviews of Modern Physics**, v. 74, n. 1, p. 47–97, January 2002.

AMINI, A.; WAH, T. Y.; TEH, Y. W. DENGRIS-Stream: a density-grid based clustering algorithm for evolving data streams over sliding window. In: **Proceedings of the International Conference on Data Mining and Computer Engineering**. Bangkok, Thailand: Planetary Scientific Research Center (PSRC), 2012. p. 206–210.

ARCHAMBAULT, D.; PURCHASE, H. C. Can animation support the visualisation of dynamic graphs? **Information Sciences**, v. 330, p. 495 – 509, 2016. ISSN 0020-0255. Disponível em: <https://doi.org/10.1016/j.ins.2015.04.017>.

BABCOCK, B.; BABU, S.; DATAR, M.; MOTWANI, R.; WIDOM, J. Models and issues in data stream systems. In: **Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems**. New York, NY, United States: Association for Computing Machinery, 2002. p. 1–16. Disponível em: <https://doi.org/10.1145/543613.543615>.

BABU, S.; WIDOM, J. Continuous queries over data streams. **ACM Sigmod Record**, ACM, v. 30, n. 3, p. 109–120, 2001. Disponível em: <https://doi.org/10.1145/603867.603884>.

BACH, B. Unfolding dynamic networks for visual exploration. **IEEE Computer Graphics and Applications**, v. 36, n. 2, p. 74–82, Mar 2016. ISSN 0272-1716. Disponível em: <ttps://doi.org/10.1109/MCG.2016.32>.

BACH, B.; HENRY-RICHE, N.; DWYER, T.; MADHYASTHA, T.; FEKETE, J.-D.; GRABOWSKI, T. Small multipiles: Piling time to explore temporal patterns in dynamic networks. **Computer Graphics Forum**, v. 34, n. 3, p. 31–40, 2015. Disponível em: <https://doi.org/10.1111/cgf.12615>.

BACH, B.; PIETRIGA, E.; FEKETE, J.-D. GraphDiaries: Animated Transitions and Temporal Navigation for Dynamic Networks. **IEEE Transactions on Visualization and Computer Graphics**, Institute of Electrical and Electronics Engineers, v. 20, n. 5, p. 740 – 754, maio 2014. Disponível em: <https://doi.org/10.1109/TVCG.2013.254>.

BARRAT, A.; BARTHÉLEMY, M.; VESPIGNANI, A. **Dynamical Processes on Complex Networks**. Cambridge University Press, 2008. Disponível em: <https://doi.org/10.1017/CBO9780511791383>.

BASAILLE, I.; KIRGIZOV, S.; LECLERCQ, ; SAVONNET, M.; CULLOT, N. Towards a twitter observatory: A multi-paradigm framework for collecting, storing and analysing tweets. In: **2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)**. Grenoble, France: [s.n.], 2016. p. 1–10. Disponível em: <https://doi.org/10.1109/RCIS.2016.7549324>.

BASTIAN, M.; HEYMANN, S.; JACOMY, M. Gephi: An open source software for exploring and manipulating networks. In: **International AAAI Conference on Weblogs and Social Media**. [s.n.], 2009. Disponível em: <https://doi.org/10.13140/2.1.1341.1520>.

BATTISTA, G. D.; EADES, P.; TAMASSIA, R.; TOLLIS, I. G. Algorithms for drawing graphs: an annotated bibliography. **Computational Geometry**, v. 4, n. 5, p. 235 – 282, 1994. Disponível em: <https://doi.org/10.1016/0925-7721(94)00014-X>.

BECK, F.; BURCH, M.; DIEHL, S.; WEISKOPF, D. A taxonomy and survey of dynamic graph visualization. **Computer Graphics Forum**, The Eurographs Association & John Wiley & Sons, Ltd., Chichester, UK, v. 36, n. 1, p. 133–159, January 2017. ISSN 0167-7055. Disponível em: <https://doi.org/10.1111/cgf.12791>.

BEHRISCH, M.; BACH, B.; RICHE, N. H.; SCHRECK, T.; FEKETE, J.-D. Matrix reordering methods for table and network visualization. In: WILEY ONLINE LIBRARY. **Computer Graphics Forum**. 2016. v. 35, n. 3, p. 693–716. Disponível em: <https://doi.org/10.1111/cgf.12935>.

BLONDEL, V. D.; GUILLAUME, J.-L.; LAMBIOTTE, R.; LEFEBVRE, E. Fast unfolding of communities in large networks. **Journal of Statistical Mechanics: Theory and Experiment**, v. 2008, n. 10, p. P10008, October 2008. Disponível em: <https://doi.org/10.1088/1742-5468/2008/10/p10008>.

BONACCORSI, S.; OTTAVIANO, S.; PELLEGRINI, F. D.; SOCIEVOLE, A.; MIEGHEM, P. V. Epidemic outbreaks in two-scale community networks. **Physical Review E**, American Physical Society, v. 90, n. 1, p. 012810, 2014. Disponível em: <https://doi.org/10.1103/PhysRevE.90.012810>.

BRAVERMAN, V.; OSTROVSKY, R.; ZANIOLO, C. Optimal sampling from sliding windows. In: **Proceedings of the Twenty-Eighth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems**. New York, NY, USA: Association for Computing Machinery, 2009. (PODS '09), p. 147–156. ISBN 9781605585536. Disponível em: <https://doi.org/10.1145/1559795.1559818>.

BU, Z.; WANG, Y.; LI, H.-J.; JIANG, J.; WU, Z.; CAO, J. Link prediction in temporal networks: Integrating survival analysis and game theory. **Information Sciences**, v. 498, p. 41 – 61, 2019. ISSN 0020-0255. Disponível em: <https://doi.org/10.1016/j.ins.2019.05.050>.

BURCH, M. Visual analytics of large dynamic digraphs. **Information Visualization**, v. 16, n. 3, p. 167–178, 2017. Disponível em: <https://doi.org/10.1177/1473871616661194>.

BURCH, M.; WEISKOPF, D. A flip-book of edge-splatted small multiples for visualizing dynamic graphs. In: **Proceedings of the 7th International Symposium on Visual Information Communication and Interaction**. New York, NY, USA: Association for Computing Machinery, 2014. (VINCI '14), p. 29–38. ISBN 9781450327657. Disponível em: <https://doi.org/10.1145/2636240.2636839>.

CAO, F.; ESTERT, M.; QIAN, W.; ZHOU, A. Density-based clustering over an evolving data stream with noise. In: **Proceedings of the 2006 SIAM International**

Conference on Data Mining. Society for Industrial and Applied Mathematics, 2006. p. 328–339. Disponível em: <https://doi.org/10.1137/1.9781611972764.29>.

CARD, S.; MACKINLAY, J.; SHNEIDERMAN, B. Information visualization. **Human-computer interaction: Design issues, solutions, and applications**, Taylor & Francis, v. 181, 2009.

CARD, S. K.; MACKINLAY, J. D.; SHNEIDERMAN, B. **Readings in information visualization: using vision to think**. San Francisco, CA, United States: Morgan Kaufmann, 1999. ISBN 978-1-55860-533-6.

CARRASCOSA, J. M.; GONZÁLEZ, R.; CUEVAS, R.; AZCORRA, A. Are trending topics useful for marketing?: Visibility of trending topics vs traditional advertisement. In: **Proceedings of the First ACM Conference on Online Social Networks**. New York, NY, USA: ACM, 2013. (COSN '13), p. 165–176. ISBN 978-1-4503-2084-9. Disponível em: <http://doi.acm.org/10.1145/2512938.2512948>.

CATTUTO, C.; BROECK, W. Van den; BARRAT, A.; COLIZZA, V.; PINTON, J.-F.; VESPIGNANI, A. Dynamics of person-to-person interactions from distributed rfid sensor networks. **PloS one**, Public Library of Science, v. 5, n. 7, p. e11596, 2010. Disponível em: <https://doi.org/10.1371/journal.pone.0011596>.

CAZABET, R.; ROSSETTI, G. Challenges in community discovery on temporal networks. In: _____. **Temporal Network Theory**. Cham: Springer International Publishing, 2019. p. 181–197. ISBN 978-3-030-23495-9. Disponível em: <https://doi.org/10.1007/978-3-030-23495-9_10>.

CHEN, C. Information visualization. **Wiley Interdisciplinary Reviews: Computational Statistics**, Wiley Online Library, v. 2, n. 4, p. 387–403, 2010.

COL, A. D.; VALDIVIA, P.; PETRONETTO, F.; DIAS, F.; SILVA, C. T.; NONATO, L. G. Wavelet-based visual analysis of dynamic networks. **IEEE Transactions on Visualization and Computer Graphics**, v. 24, n. 8, p. 2456–2469, 2018.

COSTA, L. da F.; JR., O. N. O.; TRAVIESO, G.; RODRIGUES, F. A.; BOAS, P. R. V.; ANTIQUEIRA, L.; VIANA, M. P.; ROCHA, L. E. C. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. **Advances in Physics**, Taylor & Francis, v. 60, n. 3, p. 329–412, 2011. Disponível em: <https://doi.org/10.1080/00018732.2011.572452>.

CRAMPES, M.; PLANTIÉ, M. A unified community detection, visualization and analysis method. **Advances in Complex Systems**, v. 17, 2014. Disponível em: <https://doi.org/10.1142/S0219525914500015>.

CRNOVRSANIN, T.; CHU, J.; MA, K.-L. An incremental layout method for visualizing online dynamic graphs. In: GIACOMO, E. D.; LUBIW, A. (Ed.). **Graph Drawing and Network Visualization**. Cham: Springer International Publishing, 2015. p. 16–29. ISBN 978-3-319-27261-0. Disponível em: <https://doi.org/10.1007/978-3-319-27261-0_2>.

DIAS, M. D.; MANSOUR, M. R.; DIAS, F.; PETRONETTO, F.; SILVA, C. T.; NONATO, L. G. A hierarchical network simplification via non-negative matrix factorization. In: **2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.: s.n.], 2017. p. 119–126.

DRIF, A.; BOUKERRAM, A. Taxonomy and survey of community discovery methods in complex networks. **International Journal of Computer Science and Engineering Survey**, Academy & Industry Research Collaboration Center (AIRCC), v. 5, n. 4, p. 1, 2014. Disponível em: <https://doi.org/10.5121/ijcses.2014.5401>.

ELLIS, G.; DIX, A. A taxonomy of clutter reduction for information visualisation. **IEEE Transactions on Visualization and Computer Graphics**, v. 13, n. 6, p. 1216–1223, 2007. Disponível em: <https://doi.org/10.1109/TVCG.2007.70535>.

ELZEN, S. van den; HOLTEN, D.; BLAAS, J.; WIJK, J. J. van. Reordering massive sequence views: Enabling temporal and structural analysis of dynamic networks. In: **2013 IEEE Pacific Visualization Symposium (PacificVis)**. IEEE, 2013. p. 33–40. ISSN 2165-8765. Disponível em: <https://doi.org/10.1109/PacificVis.2013.6596125>.

_____. Dynamic network visualization with extended massive sequence views. **IEEE Transactions on Visualization and Computer Graphics**, IEEE, v. 20, n. 8, p. 1087–1099, Aug 2014. ISSN 1077-2626. Disponível em: <https://doi.org/10.1109/TVCG.2013.263>.

_____. Reducing snapshots to points: A visual analytics approach to dynamic network exploration. **IEEE Transactions on Visualization and Computer Graphics**, IEEE, v. 22, n. 1, p. 1–10, Jan 2016. ISSN 1077-2626. Disponível em: <https://doi.org/10.1109/TVCG.2015.2468078>.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: **Proceedings of the Second International Conference on Knowledge Discovery and Data Mining**. [S.l.]: AAAI Press, 1996. (KDD'96), p. 226–231.

ESTRADA, E. Communicability in temporal networks. **Physical Review E**, APS, v. 88, n. 4, p. 042811, 2013.

_____. Introduction to complex networks: Structure and dynamics. In: _____. **Evolutionary Equations with Applications in Natural Sciences**. Cham: Springer International Publishing, 2015. p. 93–131. ISBN 978-3-319-11322-7. Disponível em: <http://dx.doi.org/10.1007/978-3-319-11322-7_3>.

ETEMADI, R.; LU, J. Pes: Priority edge sampling in streaming triangle estimation. **IEEE Transactions on Big Data**, IEEE, 2019. Disponível em: <https://doi.org/10.1109/TBDATA.2019.2948613>.

FORTUNATO, S. Community detection in graphs. **Physics Reports**, v. 486, n. 3, p. 75 – 174, 2010. ISSN 0370-1573.

FORTUNATO, S.; BARTHÉLEMY, M. Resolution limit in community detection. **Proceedings of the National Academy of Sciences**, v. 104, n. 1, p. 36–41, 2007. Disponível em: <https://doi.org/10.1073/pnas.0605965104>.

FORTUNATO, S.; HRIC, D. Community detection in networks: A user guide. **Physics Reports**, v. 659, n. Supplement C, p. 1 – 44, 2016. ISSN 0370-1573. Disponível em: <https://doi.org/10.1016/j.physrep.2016.09.002>.

FRISHMAN, Y.; TAL, A. Online dynamic graph drawing. **IEEE Transactions on Visualization and Computer Graphics**, IEEE, v. 14, n. 4, p. 727–740, July 2008. ISSN 2160-9306. Disponível em: <https://doi.org/10.1109/TVCG.2008.11>.

GAMA, J. **Knowledge Discovery from Data Streams**. 1st. ed. [S.l.]: Chapman & Hall/CRC, 2010. ISBN 1439826110, 9781439826119.

GAMA, J.; SEBASTIÃO, R.; RODRIGUES, P. P. On evaluating stream learning algorithms. **Machine Learning**, v. 90, n. 3, p. 317–346, Mar 2013. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/s10994-012-5320-9>.

GEMMETTO, V.; BARRAT, A.; CATTUTO, C. Mitigation of infectious disease at school: targeted class closure vs school closure. **BMC infectious diseases**, BioMed Central Ltd, v. 14, n. 1, p. 695, dez. 2014. ISSN 1471-2334. Disponível em: <https://doi.org/10.1186/s12879-014-0695-9>.

GHONIEM, M.; SHURKHOVETSKYY, G.; BAHEY, A.; OTJACQUES, B. VAFLE: Visual analytics of firewall log events. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Visualization and Data Analysis 2014**. 2014. v. 9017, p. 901704. Disponível em: <https://doi.org/10.1117/12.2037790>.

GIALAMPOUKIDIS, I.; TSIKRIKA, T.; VROCHIDIS, S.; KOMPATSIARIS, I. Community detection in complex networks based on dbscan* and a martingale process. In: IEEE. **Semantic and Social Media Adaptation and Personalization (SMAP), 2016 11th International Workshop on**. 2016. p. 1–6. Disponível em: <https://doi.org/10.1109/SMAP.2016.7753375>.

GOROCHOWSKI, T. E.; BERNARDO, M. di; GRIERSON, C. S. Using aging to visually uncover evolutionary processes on networks. **IEEE Transactions on Visualization and Computer Graphics**, v. 18, n. 8, p. 1343–1352, Aug 2012. ISSN 2160-9306. Disponível em: <https://doi.org/10.1109/TVCG.2011.142>.

GRABOWICZ, P. A.; AIELLO, L. M.; MENCZER, F. Fast filtering and animation of large dynamic networks. **EPJ Data Science**, SpringerOpen, v. 3, n. 1, p. 27, 2014. Disponível em: <https://doi.org/10.1140/epjds/s13688-014-0027-8>.

GREENWALD, M.; KHANNA, S. et al. Space-efficient online computation of quantile summaries. **ACM SIGMOD Record**, v. 30, n. 2, p. 58–66, 2001. Disponível em: <https://doi.org/10.1145/376284.375670>.

GUIDOTTI, R.; COSCIA, M. On the equivalence between community discovery and clustering. In: SPRINGER. **International Conference on Smart Objects and Technologies for Social Good**. 2017. p. 342–352. Disponível em: <https://doi.org/10.1007/978-3-319-76111-4_34>.

HAN, J.; KAMBER, M. **Data Mining Concepts and Techniques (2nd Edition)**. [S.l.]: Morgan Kaufmann, 2006. ISBN 1-55860-901-6.

HOLME, P.; LILJEROS, F. Birth and death of links control disease spreading in empirical contact networks. **Scientific Reports**, v. 4, n. 4999, 2014. Disponível em: <https://doi.org/10.1038/srep04999>.

HOLME, P.; SARAMÄKI, J. Temporal networks. **Physics Reports**, Elsevier, v. 519, n. 3, p. 97–125, October 2012.

HOLTEN, D.; CORNELISSEN, B.; WIJK, J. J. van. Trace visualization using hierarchical edge bundles and massive sequence views. In: **2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis**. IEEE, 2007. p. 47–54. Disponível em: <https://doi.org/10.1109/VISSOF.2007.4290699>.

HOLTEN, D.; WIJK, J. J. V. Force-directed edge bundling for graph visualization. **Computer Graphics Forum**, v. 28, n. 3, p. 983–990, 2009. Disponível em: <https://doi.org/10.1111/j.1467-8659.2009.01450.x>.

IMRAN, M.; CASTILLO, C.; LUCAS, J.; MEIER, P.; VIEWEG, S. Aidr: Artificial intelligence for disaster response. In: **Proceedings of the 23rd International Conference on World Wide Web**. New York, NY, USA: ACM, 2014. (WWW '14 Companion), p. 159–162. ISBN 978-1-4503-2745-9. Disponível em: <https://doi.org/10.1145/2567948.2577034>.

ISELLA, L.; STEHLÉ, J.; BARRAT, A.; CATTUTO, C.; PINTON, J.-F.; BROECK, W. V. den. What's in a crowd? analysis of face-to-face behavioral networks. **Journal of Theoretical Biology**, v. 271, p. 166–180, 2011. Disponível em: <https://doi.org/10.1016/j.jtbi.2010.11.033>.

JHA, M.; SESHADHRI, C.; PINAR, A. A space-efficient streaming algorithm for estimating transitivity and triangle counts using the birthday paradox. **ACM Transactions on Knowledge Discovery from Data**, Association for Computing Machinery, New York, NY, USA, v. 9, n. 3, fev. 2015. ISSN 1556-4681. Disponível em: <https://doi.org/10.1145/2700395>.

JO, H.-H.; HIRAOKA, T. Bursty time series analysis for temporal networks. In: _____. **Temporal Network Theory**. Cham: Springer International Publishing, 2019. p. 161–179. ISBN 978-3-030-23495-9. Disponível em: <https://doi.org/10.1007/978-3-030-23495-9_9>.

JRAD, N.; KACHENOURA, A.; NICA, A.; MERLET, I.; WENDLING, F. A Page-Hinkley based method for HFOs detection in epileptic depth-EEG. In: IEEE. **2017 25th European Signal Processing Conference (EUSIPCO)**. 2017. p. 1295–1299. Disponível em: <https://doi.org/10.23919/EUSIPCO.2017.8081417>.

KAPANIPATHI, P.; JAIN, P.; VENKATARAMANI, C.; SHETH, A. Hierarchical interest graph from tweets. In: **Proceedings of the 23rd International Conference on World Wide Web**. New York, NY, USA: ACM, 2014. (WWW '14 Companion), p. 311–312. ISBN 978-1-4503-2745-9. Disponível em: <https://doi.org/10.1145/2567948.2577353>.

KEILA, P. S.; SKILLICORN, D. B. Structure in the enron email dataset. **Computational & Mathematical Organization Theory**, Kluwer Academic Publishers, Hingham, MA, USA, v. 11, n. 3, p. 183–199, out. 2005. ISSN 1381-298X. Disponível em: <https://doi.org/10.1007/s10588-005-5379-y>.

KEIM, D. A. Information visualization and visual data mining. **IEEE Transactions on Visualization and Computer Graphics**, IEEE Educational Activities

Department, USA, v. 8, n. 1, p. 1–8, jan. 2002. ISSN 1077-2626. Disponível em: <https://doi.org/10.1109/2945.981847>.

KRANEN, P.; ASSENT, I.; BALDAUF, C.; SEIDL, T. The ClusTree: indexing micro-clusters for anytime stream mining. **Knowledge and Information Systems**, Springer, v. 29, n. 2, p. 249–272, 2011. Disponível em: <https://doi.org/10.1007/s10115-010-0342-8>.

KRSTAJIĆ, M.; KEIM, D. A. Visualization of streaming data: Observing change and context in information visualization techniques. In: **2013 IEEE International Conference on Big Data**. IEEE, 2013. p. 41–47. ISBN 978-1-4799-1293-3. Disponível em: <https://doi.org/10.1109/BigData.2013.6691713>.

KYROLA, A.; BLELLOCH, G.; GUESTRIN, C. Graphchi: Large-scale graph computation on just a pc. In: **Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation**. Berkeley, CA, USA: USENIX Association, 2012. (OSDI'12), p. 31–46. ISBN 978-1-931971-96-6. Disponível em: <https://doi.org/10.21236/ada603410>.

LAMBERT, A.; BOURQUI, R.; AUBER, D. 3d edge bundling for geographical data visualization. In: **2010 14th International Conference Information Visualisation**. IEEE, 2010. p. 329–335. ISSN 2375-0138. Disponível em: <https://doi.org/10.1109/IV.2010.53>.

LANCICHINETTI, A.; FORTUNATO, S. Community detection algorithms: A comparative analysis. **Physical Review E**, American Physical Society, v. 80, p. 056117, Nov 2009. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevE.80.056117>.

LEE, E.; MOODY, J.; MUCHA, P. J. Exploring concurrency and reachability in the presence of high temporal resolution. In: _____. **Temporal Network Theory**. Cham: Springer International Publishing, 2019. p. 129–145. ISBN 978-3-030-23495-9. Disponível em: <https://doi.org/10.1007/978-3-030-23495-9_7>.

LHUILLIER, A.; HURTER, C.; TELEA, A. FFTEB: Edge bundling of huge graphs by the fast fourier transform. In: **2017 IEEE Pacific Visualization Symposium (PacificVis)**. IEEE, 2017. p. 190–199. ISSN 2165-8773. Disponível em: <https://doi.org/10.1109/PACIFICVIS.2017.8031594>.

LI, M.; YANG, J.; WU, F.-X.; PAN, Y.; WANG, J. DyNetViewer: a Cytoscape app for dynamic network construction, analysis and visualization. **Bioinformatics**, v. 34, n. 9, p. 1597–1599, 12 2017. ISSN 1367-4803. Disponível em: <https://doi.org/10.1093/bioinformatics/btx821>.

LIM, K. H.; DATTA, A. Following the follower: Detecting communities with common interests on twitter. In: **Proceedings of the 23rd ACM Conference on Hypertext and Social Media**. New York, NY, USA: ACM, 2012. (HT '12), p. 317–318. ISBN 978-1-4503-1335-3. Disponível em: <https://doi.org/10.1145/2309996.2310052>.

LIN, C.-C.; LEE, Y.-Y.; YEN, H.-C. Mental map preserving graph drawing using simulated annealing. **Information Sciences**, Elsevier, v. 181, n. 19, p. 4253–4272, 2011. Disponível em: <https://doi.org/10.1016/j.ins.2011.06.005>.

LINHARES, C. D. G.; PONCIANO, J. R.; PAIVA, J. G. S.; TRAVENÇOLO, B. A. N.; ROCHA, L. E. C. Visualisation of structure and processes on temporal networks. In: _____. **Temporal Network Theory**. Cham: Springer International Publishing, 2019. p. 83–105. ISBN 978-3-030-23495-9. Disponível em: <https://doi.org/10.1007/978-3-030-23495-9_5>.

LINHARES, C. D. G.; PONCIANO, J. R.; PEREIRA, F. S. F.; ROCHA, L. E. C.; PAIVA, J. G. S.; TRAVENÇOLO, B. A. N. A scalable node ordering strategy based on community structure for enhanced temporal network visualization. **Computers & Graphics**, v. 84, p. 185 – 198, 2019. ISSN 0097-8493. Disponível em: <https://doi.org/10.1016/j.cag.2019.08.006>.

_____. Visual analysis for evaluation of community detection algorithms. **Multimedia Tools and Applications**, Springer, v. 79, n. 25, p. 17645–17667, 2020. Disponível em: <https://doi.org/10.1007/s11042-020-08700-4>.

LINHARES, C. D. G.; PONCIANO, J. R.; TRAVENÇOLO, B. A. N.; PAIVA, J. G. S.; ROCHA, L. E. C. Análise temporal de uma rede de contato hospitalar utilizando técnicas de visualização de informação. In: . XVII Workshop de Informática Médica, 2017. Disponível em: <https://doi.org/10.5753/sbcas.2017.3696>.

LINHARES, C. D. G.; TRAVENÇOLO, B. A. N.; PAIVA, J. G. S.; ROCHA, L. E. C. DyNetVis: A system for visualization of dynamic networks. In: **Proceedings of the Symposium on Applied Computing**. Marrakech, Morocco: ACM, 2017. (SAC '17), p. 187–194. ISBN 978-1-4503-4486-9. Disponível em: <https://doi.org/10.1145/3019612.3019686>.

LIU, X.; GUAN, J.; HU, P. Mining frequent closed itemsets from a landmark window over online data streams. **Computers & Mathematics with Applications**, Pergamon Press, Inc., USA, v. 57, n. 6, p. 927–936, mar. 2009. ISSN 0898-1221. Disponível em: <https://doi.org/10.1016/j.camwa.2008.10.060>.

MALEWICZ, G.; AUSTERN, M. H.; BIK, A. J.; DEHNERT, J. C.; HORN, I.; LEISER, N.; CZAJKOWSKI, G. Pregel: A system for large-scale graph processing. In: **Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data**. New York, NY, USA: ACM, 2010. (SIGMOD '10), p. 135–146. ISBN 978-1-4503-0032-2. Disponível em: <https://doi.org/10.1145/1807167.1807184>.

MANKU, G. S.; MOTWANI, R. Approximate frequency counts over data streams. In: **Proceedings of the 28th International Conference on Very Large Data Bases**. [S.l.]: VLDB Endowment, 2002. (VLDB '02), p. 346–357.

MANSMANN, F.; KRSTAJIC, M.; FISCHER, F.; BERTINI, E. StreamSqueeze: a dynamic stream visualization for monitoring of event data. In: **Visualization and Data Analysis**. [s.n.], 2012. p. 829404. Disponível em: <https://doi.org/10.1117/12.912372>.

MASTRANDREA, R.; FOURNET, J.; BARRAT, A. Contact patterns in a high school: A comparison between data collected using wearable sensors, contact diaries and friendship surveys. **PLOS ONE**, Public Library of Science, v. 10, n. 9, p. 1–26, 09 2015. Disponível em: <https://doi.org/10.1371/journal.pone.0136497>.

MASUD, M.; GAO, J.; KHAN, L.; HAN, J.; THURAISINGHAM, B. M. Classification and novel class detection in concept-drifting data streams under time constraints. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 23, n. 6, p. 859–874, 2011. Disponível em: <https://doi.org/10.1109/TKDE.2010.61>.

MASUDA, N.; LAMBIOTTE, R. **A Guide to Temporal Networks**. World Scientific, 2016. Disponível em: <https://doi.org/10.1142/q0033>.

MCGREGOR, A. Graph mining on streams. In: _____. **Encyclopedia of Database Systems**. Boston, MA: Springer US, 2009. p. 1271–1275. ISBN 978-0-387-39940-9. Disponível em: <https://doi.org/10.1007/978-0-387-39940-9_184>.

METWALLY, A.; AGRAWAL, D.; ABBADI, A. E. Efficient computation of frequent and top-k elements in data streams. In: SPRINGER. **International Conference on Database Theory**. 2005. p. 398–412. Disponível em: <https://doi.org/10.1007/978-3-540-30570-5_27>.

MI, P.; SUN, M.; MASIANE, M.; CAO, Y.; NORTH, C. Interactive graph layout of a million nodes. **Informatics**, v. 3, p. 23, 2016. Disponível em: <https://doi.org/10.3390/informatics3040023>.

MITRA, B.; TABOURIER, L.; ROTH, C. Intrinsically dynamic network communities. **Computer Networks**, v. 56, n. 3, p. 1041–1053, 2012. Disponível em: <https://doi.org/10.1016/j.comnet.2011.10.024>.

MOTHE, J.; MKHITARYAN, K.; HAROUTUNIAN, M. Community detection: Comparison of state of the art algorithms. In: **2017 Computer Science and Information Technologies (CSIT)**. IEEE, 2017. p. 125–129. Disponível em: <https://doi.org/10.1109/CSITechnol.2017.8312155>.

NEWMAN, M. **Networks: An Introduction**. USA: Oxford University Press, Inc., 2010. ISBN 0199206651.

NONATO, L. G.; CARMO, F. P.; SILVA, C. T. GLoG: Laplacian of gaussian for spatial pattern detection in spatio-temporal data. **IEEE Transactions on Visualization and Computer Graphics**, p. 1–1, 2020.

ORMAN, G. K.; CHERIFI, H.; LABATUT, V. On accuracy of community structure discovery algorithms. **Journal of Convergence Information Technology**, v. 6, p. 283–292, 2011. Disponível em: <https://doi.org/10.4156/jcit.vol6.issue11.32>.

ORMAN, G. K.; LABATUT, V.; CHERIFI, H. Comparative evaluation of community detection algorithms: a topological approach. **Journal of Statistical Mechanics: Theory and Experiment**, IOP Publishing, v. 2012, n. 08, p. P08001, aug 2012. Disponível em: <https://doi.org/10.1088%2F1742-5468%2F2012%2F08%2Fp08001>.

ORMAN, G. K.; LABATUT, V.; PLANTEVIT, M.; BOULICAUT, J. F. A method for characterizing communities in dynamic attributed complex networks. In: **2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)**. IEEE, 2014. p. 481–484. Disponível em: <https://doi.org/10.1109/ASONAM.2014.6921629>.

PAGE, E. S. Continuous inspection schemes. **Biometrika**, JSTOR, v. 41, n. 1/2, p. 100–115, 1954. Disponível em: <https://doi.org/10.2307/2333009>.

PAIVA, E. R. d. F. **Detecção de novidade em fluxos contínuos de dados multiclasse**. Tese (Doutorado) — Universidade de São Paulo, 2014.

PEREIRA, F. S.; AMO, S. d.; GAMA, J. Detecting events in evolving social networks through node centrality analysis. In: **Workshop on Large-scale Learning from Data Streams in Evolving Environments of ECML PKDD**. [S.l.: s.n.], 2016. p. 83–93.

PEREIRA, F. S. F.; AMO, S. d.; GAMA, J. Evolving centralities in temporal graphs: A twitter network analysis. In: **2016 17th IEEE International Conference on Mobile Data Management (MDM)**. IEEE, 2016. v. 2, p. 43–48. Disponível em: <https://doi.org/10.1109/MDM.2016.88>.

PEREIRA, F. S. F. et al. **User preference dynamics on evolving social networks: learning, modeling and prediction**. Tese (Doutorado) — Universidade Federal de Uberlândia, 2018.

PERER, A.; SHNEIDERMAN, B. Integrating statistics and visualization: Case studies of gaining clarity during exploratory data analysis. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2008. (CHI '08), p. 265–274. ISBN 978-1-60558-011-1. Disponível em: <http://doi.acm.org/10.1145/1357054.1357101>.

PONCIANO, J. R.; LINHARES, C. D. G.; MELO, S. L.; LIMA, L. V.; TRAVENCOLO, B. A. N. Visual analysis of contact patterns in school environments. **Informatics in Education**, Vilnius University Institute of Data Science and Digital Technologies, v. 19, n. 3, p. 455–472, 2020. ISSN 1648-5831. Disponível em: <https://doi.org/10.15388/infedu.2020.20>.

PORTER, M. A.; ONNELA, J.-P.; MUCHA, P. J. Communities in networks. **Notices of the American Mathematical Society**, v. 56, n. 9, p. 1082–1097, March 2009.

PURCHASE, H. C.; SAMRA, A. Extremes are better: Investigating mental map preservation in dynamic graphs. In: **International Conference on Theory and Application of Diagrams**. Springer, Berlin, Heidelberg, 2008. p. 60–73. Disponível em: <https://doi.org/10.1007/978-3-540-87730-1_9>.

RANSHOUS, S.; SHEN, S.; KOUTRA, D.; HARENBERG, S.; FALOUTSOS, C.; SAMATOVA, N. F. Anomaly detection in dynamic networks: A survey. **WIREs Computational Statistics**, John Wiley & Sons, Inc., New York, NY, USA, v. 7, n. 3, p. 223–247, maio 2015. ISSN 1939-5108. Disponível em: <https://doi.org/10.1002/wics.1347>.

REN, J.; MA, R. Density-based data streams clustering over sliding windows. In: IEEE. **2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery**. IEEE, 2009. v. 5, p. 248–252. Disponível em: <https://doi.org/10.1109/FSKD.2009.553>.

REY, G. D.; DIEHL, S. Controlling presentation speed, labels, and tooltips in interactive animations. **Journal of Media Psychology**, Hogrefe Publishing, 2010. Disponível em: <https://doi.org/10.1027/1864-1105/a000021>.

ROCHA, L. E. C.; BLONDEL, V. D. Bursts of vertex activation and epidemics in evolving networks. **PLOS Computational Biology**, Public Library of Science, v. 9, p. 1–9, 2013. Disponível em: <https://doi.org/10.1371/journal.pcbi.1002974>.

ROCHA, L. E. C.; LILJEROS, F.; HOLME, P. Information dynamics shape the sexual networks of internet-mediated prostitution. **Proceedings of the National Academy of Sciences**, National Academy of Sciences, v. 107, n. 13, p. 5706–5711, 2010. ISSN 0027-8424. Disponível em: <https://doi.org/10.1073/pnas.0914080107>.

_____. Simulated epidemics in an empirical spatiotemporal network of 50,185 sexual contacts. **PLoS Comput Biol**, Public Library of Science, v. 7, n. 3, p. e1001109, 03 2011. Disponível em: <http://dx.doi.org/10.1371%2Fjournal.pcbi.1001109>.

ROCHA, L. E. C.; MASUDA, N.; HOLME, P. Sampling of temporal networks: Methods and biases. **Phys. Rev. E**, American Physical Society, v. 96, p. 052302, Nov 2017. Disponível em: <https://doi.org/10.1103/PhysRevE.96.052302>.

ROSVALL, M.; BERGSTROM, C. T. Maps of random walks on complex networks reveal community structure. **Proceedings of the National Academy of Sciences**, v. 105, n. 4, p. 1118–1123, 2008. Disponível em: <https://doi.org/10.1073/pnas.0706851105>.

_____. Mapping change in large networks. **PLOS ONE**, Public Library of Science, v. 5, n. 1, p. 1–7, January 2010. Disponível em: <https://doi.org/10.1371/journal.pone.0008694>.

ROSVALL, M.; DELVENNE, J.-C.; SCHAUB, M. T.; LAMBIOTTE, R. Different approaches to community detection. In: _____. **Advances in Network Clustering and Blockmodeling**. John Wiley & Sons, Ltd, 2019. cap. 4, p. 105–119. ISBN 9781119483298. Disponível em: <https://doi.org/10.1002/9781119483298.ch4>.

SAFFREY, P.; PURCHASE, H. The "mental map" versus "static aesthetic" compromise in dynamic graphs: a user study. In: **Proceedings of the ninth conference on Australasian user interface-Volume 76**. Australia: Australian Computer Society, Inc., 2008. p. 85–93. Disponível em: <https://dl.acm.org/doi/10.5555/1378337.1378354>.

SAH, P.; SINGH, L. O.; CLAUSET, A.; BANSAL, S. Exploring community structure in biological networks with random graphs. **BMC bioinformatics**, BioMed Central, v. 15, n. 1, p. 220, 2014. Disponível em: <https://doi.org/10.1186/1471-2105-15-220>.

SALLABERRY, A.; MUELDER, C.; MA, K.-L. Clustering, visualizing, and navigating for large dynamic graphs. In: DIDIMO, W.; PATRIGNANI, M. (Ed.). **Graph Drawing**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 487–498. ISBN 978-3-642-36763-2. Disponível em: <https://doi.org/10.1007/978-3-642-36763-2_43>.

SARMENTO, R.; CORDEIRO, M.; GAMA, J. Visualization for streaming telecommunications networks. In: APPICE, A.; CECI, M.; LOGLISCI, C.; MANCO, G.; MASCIARI, E.; RAS, Z. W. (Ed.). **New Frontiers in Mining Complex Patterns**.

Cham: Springer International Publishing, 2015. p. 117–131. ISBN 978-3-319-17876-9. Disponível em: <https://doi.org/10.1007/978-3-319-17876-9_8>.

SARMENTO, R.; CORDEIRO, M.; GAMA, J. a. Streaming networks sampling using top-k networks. In: **Proceedings of the 17th International Conference on Enterprise Information Systems - Volume 1**. Setubal, PRT: SCITEPRESS - Science and Technology Publications, Lda, 2015. (ICEIS 2015), p. 228–234. ISBN 9789897580963. Disponível em: <https://doi.org/10.5220/0005341402280234>.

SARMENTO, R.; OLIVEIRA, M.; CORDEIRO, M.; TABASSUM, S.; GAMA, J. Social network analysis in streaming call graphs. In: **Big Data Analysis: New Algorithms for a New Society**. Springer, 2016. p. 239–261. Disponível em: <https://doi.org/10.1007/978-3-319-26989-4_10>.

SEBASTIÃO, R.; FERNANDES, J. M. Supporting the page-hinkley test with empirical mode decomposition for change detection. In: SPRINGER. **International Symposium on Methodologies for Intelligent Systems**. Springer, Cham, 2017. p. 492–498. Disponível em: <https://doi.org/10.1007/978-3-319-60438-1_48>.

SEBASTIÃO, R.; GAMA, J. A study on change detection methods. In: **New Trends in Artificial Intelligence, 14th Portuguese Conference on Artificial Intelligence, EPIA**. [S.l.]: Springer, 2009. p. 353–364. ISBN 978-972-96895-4-3.

SEBASTIÃO, R.; SILVA, M. M.; RABIÇO, R.; GAMA, J.; MENDONÇA, T. Real-time algorithm for changes detection in depth of anesthesia signals. **Evolving Systems**, Springer, v. 4, n. 1, p. 3–12, 2013. Disponível em: <https://doi.org/10.1007/s12530-012-9063-4>.

SHANNON, P.; MARKIEL, A.; OZIER, O.; BALIGA, N. S.; WANG, J. T.; RAMAGE, D.; AMIN, N.; SCHWIKOWSKI, B.; IDEKER, T. Cytoscape: a software environment for integrated models of biomolecular interaction networks. **Genome Research**, v. 13, n. 11, p. 2498–2504, nov. 2003. Disponível em: <https://doi.org/10.1101/gr.1239303>.

SHARMA, A. Modeling the effect of people's preferences and social forces on adopting and sharing items. In: **Proceedings of the 8th ACM Conference on Recommender Systems**. New York, NY, USA: ACM, 2014. (RecSys '14), p. 421–424. ISBN 978-1-4503-2668-1. Disponível em: <https://doi.org/10.1145/2645710.2653364>.

SHETTY, J.; ADIBI, J. The enron email dataset database schema and brief statistical report. **Information sciences institute technical report, University of Southern California**, Citeseer, v. 4, n. 1, p. 120–128, 2004.

SHNEIDERMAN, B. The eyes have it: a task by data type taxonomy for information visualizations. In: **Proceedings 1996 IEEE Symposium on Visual Languages**. IEEE, 1996. p. 336–343. ISSN 1049-2615. Disponível em: <https://doi.org/10.1109/VL.1996.545307>.

SIKDAR, S.; CHAKRABORTY, T.; SARKAR, S.; GANGULY, N.; MUKHERJEE, A. ComPAS: Community preserving sampling for streaming graphs. In: INTERNATIONAL FOUNDATION FOR AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS. **Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems**. 2018. p. 184–192. Disponível em: <https://dl.acm.org/doi/10.5555/3237383.3237417>.

SIMONETTO, P.; ARCHAMBAULT, D.; KOBOUROV, S. Drawing dynamic graphs without timeslices. In: FRATI, F.; MA, K.-L. (Ed.). **Graph Drawing and Network Visualization**. Cham: Springer International Publishing, 2018. p. 394–409. ISBN 978-3-319-73915-1.

SIX, J. M.; TOLLIS, I. G. A framework and algorithms for circular drawings of graphs. **Journal of Discrete Algorithms**, v. 4, n. 1, p. 25 – 50, March 2006. ISSN 1570-8667. Disponível em: <https://doi.org/10.1016/j.jda.2005.01.009>.

STEHLÉ, J.; VOIRIN, N.; BARRAT, A.; CATTUTO, C.; ISELLA, L.; PINTON, J.; QUAGGIOTTO, M.; Van den Broeck, W.; RÉGIS, C.; LINA, B.; VANHEMS, P. High-resolution measurements of face-to-face contact patterns in a primary school. **PLOS ONE**, Public Library of Science, v. 6, n. 8, p. e23176, 08 2011. Disponível em: <http://dx.doi.org/10.1371/journal.pone.0023176>.

SUN, J.; FALOUTSOS, C.; FALOUTSOS, C.; PAPADIMITRIOU, S.; YU, P. S. Graphscope: Parameter-free mining of large time-evolving graphs. In: **Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2007. (KDD '07), p. 687–696. ISBN 978-1-59593-609-7. Disponível em: <https://doi.org/10.1145/1281192.1281266>.

TANAHASHI, Y.; MA, K. L. Design considerations for optimizing storyline visualizations. **IEEE Transactions on Visualization and Computer Graphics**, v. 18, n. 12, p. 2679–2688, Dec 2012. ISSN 1077-2626. Disponível em: <https://doi.org/10.1109/TVCG.2012.212>.

UPSON, C.; FAULHABER, T.; KAMINS, D.; LAIDLAW, D.; SCHLEGEL, D.; VROOM, J.; GURWITZ, R.; DAM, A. V. The application visualization system: A computational environment for scientific visualization. **IEEE Computer Graphics and Applications**, IEEE, v. 9, n. 4, p. 30–42, 1989. Disponível em: <https://doi.org/10.1109/38.31462>.

VANHEMS, P.; BARRAT, A.; CATTUTO, C.; PINTON, J.-F.; KHANAFER, N.; RéGIS, C.; KIM, B.-a.; COMTE, B.; VOIRIN, N. Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. **PLOS ONE**, Public Library of Science, v. 8, n. 9, p. 1–9, 09 2013. Disponível em: <https://doi.org/10.1371/journal.pone.0073970>.

VEHLOW, C.; BECK, F.; AUWÄRTER, P.; WEISKOPF, D. Visualizing the evolution of communities in dynamic graphs. **Computer Graphics Forum**, v. 34, n. 1, p. 277–288, February 2015. ISSN 1467-8659. Disponível em: <https://doi.org/10.1111/cgf.12512>.

VISWANATH, B.; MISLOVE, A.; CHA, M.; GUMMADI, K. P. On the evolution of user interaction in Facebook. In: **Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN'09)**. New York, NY, United States: Association for Computing Machinery, 2009. Disponível em: <https://doi.org/10.1145/1592665.1592675>.

VITTER, J. S. Random sampling with a reservoir. **ACM Trans. Math. Softw.**, ACM, New York, NY, USA, v. 11, n. 1, p. 37–57, mar. 1985. ISSN 0098-3500. Disponível em: <http://doi.acm.org/10.1145/3147.3165>.

WANG, W.; WANG, H.; DAI, G.; WANG, H. Visualization of large hierarchical data by circle packing. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: ACM, 2006. (CHI '06), p. 517–520. ISBN 1-59593-372-7. Disponível em: <https://doi.org/10.1145/1124772.1124851>.

WANG, Y.; ARCHAMBAULT, D.; HALEEM, H.; MOELLER, T.; WU, Y.; QU, H. Nonuniform timeslicing of dynamic graphs based on visual complexity. In: **2019 IEEE Visualization Conference (VIS)**. IEEE, 2019. p. 1–5. Disponível em: <https://doi.org/10.1109/VISUAL.2019.8933748>.

WARD, M. O.; GRINSTEIN, G.; KEIM, D. **Interactive Data Visualization: Foundations, Techniques, and Applications, Second Edition - 360 Degree Business**. 2nd. ed. USA: A. K. Peters, Ltd., 2015. ISBN 1482257378.

WARE, C. Information Visualization (third edition). In: _____. Boston: Morgan Kaufmann, 2013. (Interactive Technologies), p. 514. ISBN 978-0-12-381464-7. Disponível em: <https://doi.org/10.1016/B978-0-12-381464-7.00018-1>.

YANG, Z.; ALGESHEIMER, R.; TESSONE, C. J. A comparative analysis of community detection algorithms on artificial networks. **Scientific Reports**, v. 6, 2016. Disponível em: <https://doi.org/10.1038/srep30750>.

YIN, C.; ZHU, S.; CHEN, H.; ZHANG, B.; DAVID, B. A method for community detection of complex networks based on hierarchical clustering. **International Journal of Distributed Sensor Networks**, v. 11, n. 6, p. 849140, January 2015. Disponível em: <https://doi.org/10.1155/2015/849140>.

ZHANG, J. A survey on streaming algorithms for massive graphs. In: _____. **Managing and Mining Graph Data**. Boston, MA: Springer US, 2010. p. 393–420. ISBN 978-1-4419-6045-0. Disponível em: <https://doi.org/10.1007/978-1-4419-6045-0_13>.

ZHAO, Y.; CHEN, W.; SHE, Y.; WU, Q.; PENG, Y.; FAN, X. Visualizing dynamic network via sampled massive sequence view. In: **Proceedings of the 12th International Symposium on Visual Information Communication and Interaction**. New York, NY, USA: ACM, 2019. (VINCI'2019), p. 32:1–32:2. ISBN 978-1-4503-7626-6. Disponível em: <http://doi.acm.org/10.1145/3356422.3356454>.

ZHAO, Y.; SHE, Y.; CHEN, W.; LU, Y.; XIA, J.; CHEN, W.; LIU, J.; ZHOU, F. EOD edge sampling for visualizing dynamic network via massive sequence view. **IEEE Access**, v. 6, p. 53006–53018, 2018. ISSN 2169-3536. Disponível em: <https://doi.org/10.1109/ACCESS.2018.2870684>.